Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

a

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

a

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

Разработать приложение (консольное для простоты) которому на вход в качестве обязательного параметра передается имя файла. При старте приложение начинает считать количество бит равных 0 во всем файле. (Например для файла содержащего байты равный 0, 1 и 2 результат будет 22). Файл может быть любого размера. При этом, если запускается второй экземпляр приложения подсчет 0 бит должен производиться параллельно двумя приложениями. При старте третьего и более экземпляров приложения они должны подключаться к обработке того же файла.

Допущения:

1. Обрабатывается всегда локальный файл;

2. Файл в процессе обработки не меняется;

3. В приложение могут передаваться дополнительные параметры, если необходимо.

Требования:

1. Количество подсчитанных 0 бит одним экземпляром приложения и совокупное количество 0 бит, подсчитанное N экземплярами приложения для одного и того же файла должно совпадать;

2. Каждый байт в файле должен быть обработан только один раз, только одним экземпляром приложения;

3. Использовать только Java SE и необходимый минимум сторонних библиотек.

Уделить внимание производительности, архитектуре системы, тестируемости

a