**Описание решения**.

Документ содержит описание решения, включающее описание интерфейса, реализации и тестирования.

**Интерфейс**.

Описание интерфейса содержится в файле logger.h. Поскольку пользователь не обязан (и не должен) знать, как именно реализован класс Logger, требуемая функциональность реализуется абстрактным классом Logger. Класс Logger обладает методами показанными в таблице 1. Следующая функция возвращает указатель на экземпляр такого класса, filename представляет имя лог-файла.

std::shared\_ptr<Logger> new\_Logger(const char\* filename);

При успешном выполнении функция создает файл c заданным именем (при наличии такого файла удаляет его и создает заново). При неуспешном выполнении возвращает nullptr. При удалении объекта, на который указывает возвращаемый функцией указатель, закрывается лог-файл.

**Таблица 1**. Методы класса Logger.

|  |  |
| --- | --- |
| Метод | Пояснение |
| virtual void write(const char\* s); | Записывает в лог-файл строку s. |
| virtual Reader\* reader(); | Возвращает указатель на объект класса Reader. |
| virtual ~Logger(); | Виртуальный деструктор. |

**Примечание**. Функция write помещает данные в файл, вызывая flush, поэтому даже в том случае, если не вызывается деструктор Logger, данные попадают в файл.

Каждый поток, читающий данные из лог-файла при помощи метода reader получает указатель на объект, содержащий текущую позицию для чтения из файла, такой объект представлен абстрактным функтором Reader, для которого определен следующий оператор.

virtual bool operator()(std::string& s);

При наличии новых данных для чтения возвращает true и читает строку s, при отсутствии новых данных для чтения возвращает false.

**Реализация**.

Реализация содержится в файле logger.cpp. Схема классов показана на рис. 1, абстрактные классы показаны белым цветом, пунтирной линией показаны реализации абстрактных классов.

Reader\_Class

Reader

Logger\_Class

Logger

logger

Рис. 1. Схема классов.

Класс Logger\_Class реализует абстрактный класс Logger, функтор Reader\_Class реализует абстраткный функтор Reader.

Указатель на Logger\_Class реализован, как shared\_ptr<Logger\_Class>, при удалении всех ссылок на объект он удаляется, и файл закрывается.

При выполнении всех операций записи поток, выполняющий операцию вызывает полную блокировку объекта класса Logger\_Class, при выполнении операций чтения также выполняется полная блокировка, т.к. любая операция чтения перемещает указатель для чтения, следовательно является операцией, модифицирующей объект.

**Тестирование**.

Исходный код теста содержится в logger\_test.cpp. В тесте создаются два потока writer\_Thread и два потока reader\_Thread (см. рис. 2). Каждый поток writer\_Thread дважды с интервалом в 1 сек делает запись, содержащую метку времени. Каждому потоку reader\_Thread передается ссылка на флаг finish, пока флаг не установлен, поток делает попытки прочитать новую запись из файла при наличии такой записи читает ее и выводит в консоль. После установки флага дочитывает из файла оставшиеся записи и также выводит в консоль. В результате в лог-файл выводятся 4 записи, на консоль выводится 8 записей. Вывод в консоль имеет следующий формат.

<поток, прочитавший запись> <поток, сделавший запись> <метка времени записи>

writer1

writer2

logfile

reader1

reader2

console

Рис. 2. Схема теста.

Функция test дожидается окончания выполнения потоков writer\_Thread, после чего устанавливает флаг finish и дожидается окончания потоков reader\_Thread.