

ТЕСТОВОЕ ЗАДАНИЕ

для стажера по направлению

Разработчик C++

Цель

Разработать библиотеку для записи сообщений в журнал с разными уровнями важности и приложение, демонстрирующее работу библиотеки.

Часть 1

Разработать библиотеку для записи текстовых сообщений в журнал. В качестве журнала использовать текстовый файл.

Требования к разрабатываемой библиотеке:

- Библиотека должна иметь 2 варианта сборки: динамическая/статическая.
- Библиотека при инициализации должна принимать следующие параметры:
 - Имя файла журнала.
 - Уровень важности сообщения по умолчанию. Сообщения с уровнем ниже заданного не должны записываться в журнал. Уровень рекомендуется задавать с помощью перечисления с понятными именами. Достаточно трех уровней важности.
- В журнале должны быть сохранена следующая информация:
 - Текст сообщения.
 - Уровень важности.
 - Время получения сообщения.
- После инициализации должна быть возможность менять уровень важности сообщений по умолчанию.
- (*Дополнительно)
 - Добавить реализацию с записью лога в сокет.
 - Интерфейс логирования через сокет не должен отличаться от интерфейса логирования в файл.

Часть 2

Разработать консольное многопоточное приложение для проверки библиотеки записи сообщений в журнал.

Требования к приложению:

- Приложение должно:
 - Подключать и использовать библиотеку, реализованную в Части 1, для записи сообщений в журнал.
 - В консоли принимать сообщение и уровень важности этого сообщения от пользователя. Уровень важности может отсутствовать.
 - Передавать принятые данные от пользователя в отдельный поток для записи в журнал. Передача данных должна быть потокобезопасной.
 - Ожидать нового ввода от пользователя после передачи данных.
- Параметрами приложения должны быть имя файла журнала и уровень важности сообщения по умолчанию.
- Внутреннюю логику приложения придумать самостоятельно.

Часть 3 (*Дополнительно)

Реализовать консольную программу для сбора статистик по данным из сокета (от библиотеки логирования из части 1.5).

Требования к приложению:

- Приложение должно:
 - Принимать из сокета от библиотеки логирования данные.
 - Выводить в консоль принятое сообщение лога.
 - Выполнять подсчет статистик кол-ва сообщений:
 - сообщений всего
 - сообщений по уровню важности
 - сообщений за последний час
 - Выполнять подсчет статистик длин сообщений:
 - Минимум
 - Максимум
 - Средняя
 - Выводить в консоль собранную статистику:
 - после приема N-го сообщения
 - после таймаута T секунд, при условии, что статистика изменилась с момента последней выдачи
- Параметрами приложения должны быть параметры подключения для прослушивания сокета, значения N и T.

Требования к выполнению задания

- Программный код должен быть написан на языке C++ с использованием стандарта C++17. Следует придерживаться подходов ООП к разработке.
- Сборку проекта выполнять в среде. Для компиляции использовать компилятор g++. Цели сборки библиотеки и тестового приложения должны быть разделены.
- В папке с исходным кодом не должно быть мусора: неиспользуемых файлов исходных кодов или ресурсов, промежуточных файлов сборки и т.д.
- Не использовать никаких сторонних библиотек помимо std.
- Не использовать исключения для реализации бизнес-логики.
- В качестве целевой операционной системы, на которой будет проверяться тестовое задание, стоит рассматривать актуальные версии ubuntu/debian.
- Приложение должно корректно обрабатывать ошибки.
- Все файлы (исходный код библиотеки и приложения, в среде файлы) должны быть размещены в одном git-репозитории.

Оценка выполнения задания

Кандидат должен предоставить ссылку на git-репозиторий с выполненным заданием (части 1, 2 обязательно). Оценка будет проводиться на основе следующих критериев:

- Качество и читаемость кода.
- Корректность работы приложения с различными входными данными.
- Наличие и качество документации к выполненному заданию (README-файл, комментарии в коде и т.д.).
- Наличие и качество юнит тестирования.

Максимальное время
на выполнение
задания

1 неделя

