

# 1. Тестовое задание для стажера на направление «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

1. Библиотека должна быть динамической.
2. Библиотека при инициализации должна принимать следующие параметры:
  - a. Имя файла журнала.
  - b. 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.
3. В журнале должны быть сохранена следующая информация:
  - a. Текст сообщения.
  - b. Уровень важности.
  - c. Время получения сообщения.
4. После инициализации должна быть возможность менять уровень важности сообщений по умолчанию.

## Часть 2.

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

### Требования к приложению

1. Приложение должно:
  - a. Подключать и использовать библиотеку, реализованную в Части 1, для записи сообщений в журнал.
  - b. В консоли принимать от пользователя сообщение и его уровень важности. Уровень важности может отсутствовать.

- c. Передавать принятые данные от пользователя в отдельный поток для записи в журнал. Передача данных должна быть потокобезопасной.
  - d. Ожидать нового ввода от пользователя после передачи данных.
2. Параметрами приложения должны быть имя файла журнала и уровень важности сообщения по умолчанию.
3. Внутреннюю логику приложения придумать самостоятельно.

### **Требования к выполнению задания**

- Программный код должен быть написан на языке C++ стандарта C++17. Следует придерживаться подходов ООП к разработке.
- Сборку проекта и запуск тестов оформить в Makefile. Использовать компилятор g++ для работы в среде Linux. Цели сборки библиотеки и тестового приложения должны быть разделены. Кроме того, должна быть задана цель clean для очистки каталога от временных файлов.
- 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.
- В качестве целевой операционной системы, на которой будет проверяться тестовое задание, стоит рассматривать актуальные версии ubuntu/debian.
- Приложение должно корректно обрабатывать ошибки, связанные с записью в файлы.
- Все файлы (исходный код библиотеки и приложения, Makefile) должны быть размещены в одном git-репозитории.

### **Оценка выполнения задания**

Кандидат должен предоставить ссылку на git-репозиторий с выполненным заданием. Оценка будет проводиться на основе следующих критериев:

1. Качество и читаемость кода приложения на C++.
2. Корректность работы приложения с различными входными данными.
3. Наличие и качество документации к выполненному заданию (README-файл, комментарии в коде и т.д.).

**Максимальное время на выполнение задания:** одна неделя.