**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: Логирование, перегрузка операций**

| Студент гр. 1303 |  | Чубан Д.В. |
| --- | --- | --- |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2022

## **Цель работы.**

Реализовать класс/набор классов отслеживающих изменения состояний в программе. Отслеживание должно быть 3-х уровней:

1. Изменения состояния игрока и поля, а также срабатывание событий
2. Состояние игры (игра начата, завершена, сохранена, и.т.д.)
3. Отслеживание критических состояний и ошибок (поле инициализировано с отрицательными размерами, игрок попытался перейти на непроходимую клетку, и.т.д.)

Реализованы классы для вывода информации разных уровней для в консоль и в файл с перегруженным оператором вывода в поток

**Требования.**

* Разработан класс/набор классов отслеживающий изменения разных уровней
* Разработаны классы для вывода в консоль и файл с соблюдением идиомы RAII и перегруженным оператором вывода в поток.
* Разработанные классы спроектированы таким образом, чтобы можно было добавить новый формат вывода без изменения старого кода (например, добавить возможность отправки логов по сети)
* Выбор отслеживаемых уровней логирования должен происходить в runtime
* В runtime должен выбираться способ вывода логов (нет логирования, в консоль, в файл, в консоль и файл)

## **Описание архитектурных решений и классов.**

Для реализации системы логирования был создан класс LogPool с использованием паттерна синглтон. Данный класс хранит в себе данные о типах логируемых событий и типах вывода логов.

Для различных типов вывода были реализованы классы ConsoleLog и FileLog, которые наследуются от абстрактного класса Logger и позволяют выводить информацию в консоль и/или файл соответственно с помощью переопределенного оператора вывода *<<*.

Класс Message имеет поля с текстом лога и его типом.

Благодаря тому, что используется паттерн синглтон можно вызывать логирование любой части кода, т.к. метод LogPool getInstance(), возвращающий указатель на логгер, является статичным.

Выбор способа и уровня логирования осуществляется пользователем в начале программы. Для того чтобы создать лог достаточно создать указатель и вызвать метод getInstance(), чтобы получить доступ к логированию, после чего нужно проверить, выбрал ли пользователь данный уровень логирования с помощью метода logging(Level) и затем с помощью метода printLog(Level, string) мы создаем и записываем лог выбранным пользователем способом.

**Вывод.**

Была создана система логирования с перегрузкой операторов вывода, позволяющая записывать действия и ошибки либо в консоль, либо в текстовый файл. Создана UML-диаграмма системы логирования.