МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Объектно-ориентированное программирование»

Тема: Полиморфизм

Студент гр. 8383 _____ Мололкин К.А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Продолжить работу над игрой, изучить паттерны проектирования, необходимые для логического разделения классов.

Постановка задачи.

Разработать и реализовать набора классов для взаимодействия пользователя с юнитами и базой. Основные требования:

- Логирование действий пользователя
- Логирование действий юнитов и базы

Выполнение работы.

Для логирование были реализованы интерфейс ILogger, его наследники FileLogger, TerminalLogger, EmptyLogger, а также класс LoggerProxy.

Методы ILogger:

1. void logInformation(std::string) – логирование переданной строки;
2. void stopLogging() – функция заканчивает логирование, для класса TerminalLogger вывод всех логов в теринал;
3. void logBaseAttacked(Base* base) – логирование состояния атакованной базы, в данной функции вызывается перегруженный оператор вывода в поток;
4. void logUnitAttacked(IUnit* unit) – логирование состояния атакованной базы, в данной функции вызывается перегруженный оператор вывода в поток;

Класс FileLogger, хранит поток вывода в файл log.txt и осуществляет взаимодействие с файлом логирования по системе RAII.

Класс LoggerProxy реализует паттерн прокси для переключения между видами логирования.

Поля:

1. ILogger* logger – указатель на объект интерфейса ILogger;
2. LoggerType lType – хранит тип логгера;

методы:

1. changeLogger(LoggerType type) – смена логгера на другой тип;

Демонстрация работы.

Пример логирования сначала в файл, а потом в терминал:

```
Game board was crated with length: 12, and width: 12
Game was started!
Game board was print
Unit: Ballista was created from base 1 on possetion (6, 5)
Game board was print
Unit: Unknown type was created from base 2 on possetion (10, 10)
Game board was print
Unit on possetion: (5, 6) attacked unit on possetion: (10, 10)
Unit condition: team - 2,  health - 90, armour - 125
Unit on possetion: (5, 6) attacked base 2
Base condition: team - 2,  health - 890, num units from base - 1
```

```
Game board was print
Unit from possetinon: (10, 10) was moved to (0, 9)
Game board was print
Game was restarted
Game board was print
Game was finished
```

Выводы.

В лабораторной работе была изучена концепция полиморфизма и реализован набор классов для логирования действий пользователя и состояний программы.