

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Объектно ориентированное программирование »**  
**Тема: Полиморфизм**

Студент гр.8382

\_\_\_\_\_

Синельников М.Р

Преподаватель

\_\_\_\_\_

Жангиров Т.Р

Санкт-Петербург

2020

### **Цель работы.**

Разработать и реализовать набор классов для введения логирования действий и состояний программы.

### **Ход выполнения.**

В ходе выполнения работы были разработаны классы для взаимодействия пользователя, юнитов и базы. Общий интерфейс *Log*. От него наследуются классы *TerminalLog* и *FileLog*. Каждый класс по разному реализует виртуальную функцию *write*, отвечающую за запись логов. Для класса *TerminalLog* логи записываются в терминал, для класса *FileLog* в файл. Определение всех классов содержится в файле *Log.h*, описание в файле *Log.cpp*.

На каждом ходе пользователь может поменять тип логирования — выбрать запись в терминал, запись в файл или вообще отказаться от логирования. Переключение между разным логированием реализовано в функции *changeState* класса *Proxy*. Определение и описание в файлах *Proxy.h* и *Proxy.cpp*.

При первоначальном выборе записи в файл открывается файл *game\_report.txt*, куда и записывается всё логирование при выборе записи в файл. Открытие файла происходит в конструкторе, а закрытие в деструкторе, что соответствует идиоме RAII.

### **Выводы.**

В ходе выполнения работы были разработаны классы для логирования действий пользователя, базы и юнита.