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

ОТЧЕТ

по лабораторной работе №4

по дисциплине «ООП»

Тема: Полиморфизм

Студент гр. 8383 _____ Шишкин И.В.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Приобрести знания по полиморфизму. Реализовать логирование действий в программе.

Постановка задачи.

Реализовать набор классов, для ведения логирования действий и состояний программы. Основные требования:

Логирование действий пользователя

Логирование действий юнитов и базы

Ход работы.

Были реализованы: наборы классов для ведения логирования действий и состояний программы. Для логирования перегружен оператор вывода в поток. Для переключения между разным логированием создан паттерн «Прокси». Для разного формата записи создан паттерн «Адаптер».

Описание дополнений к программе.

Был реализован класс Logger (файлы Logger.h, Logger.cpp), который является интерфейсом для классов, используемых для логирования. Возможны 3 варианта логирования (игрок выбирает их в самом начале игры): LoggingFile – логирование в файл “log.txt” (файлы LoggingFile.h, LoggingFile.cpp); LoggingTerminal – логирование в терминал (файлы LoggingTerminal.h, LoggingTerminal.cpp); NoLogging – без логирования (файлы NoLogging.h, NoLogging.cpp). Также был перегружен оператор вывода в поток.

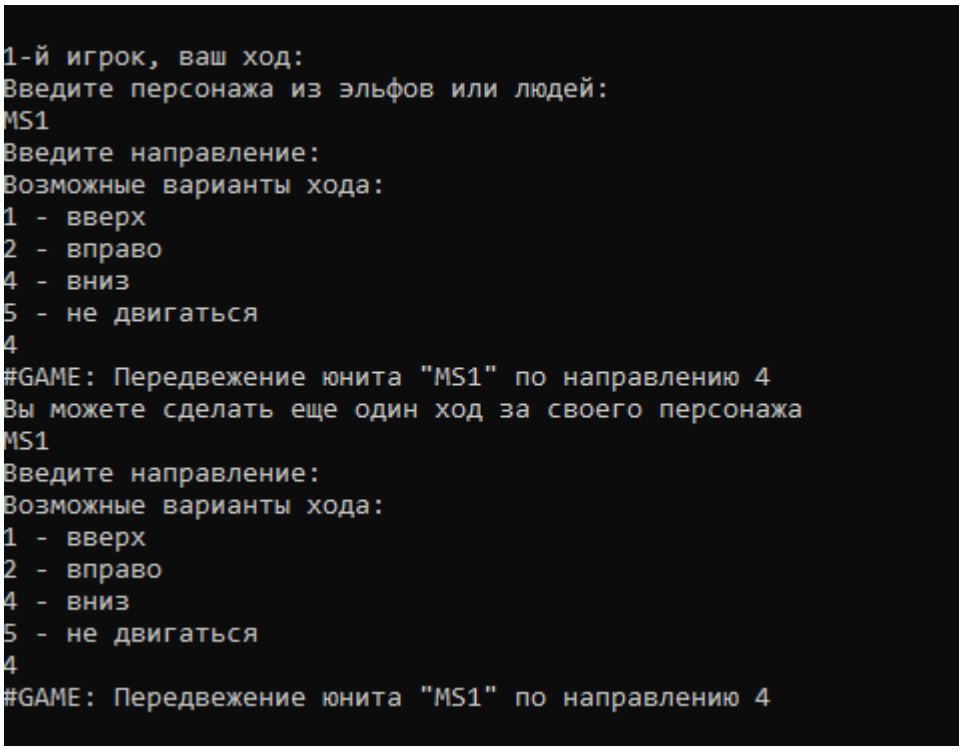
Чтобы пользователь мог переключать режимы логирования, создан класс LoggerProxy, который реализует паттерн «Прокси» (файлы LoggerProxy.h, LoggerProxy.cpp). С помощью метода setLogger можно установить другой режим логирования.

Для разных форматов вывода логов реализован класс LogFormatter (файлы LoggerFormat.h, LoggerFormat.cpp). Созданы классы FormatterTime и

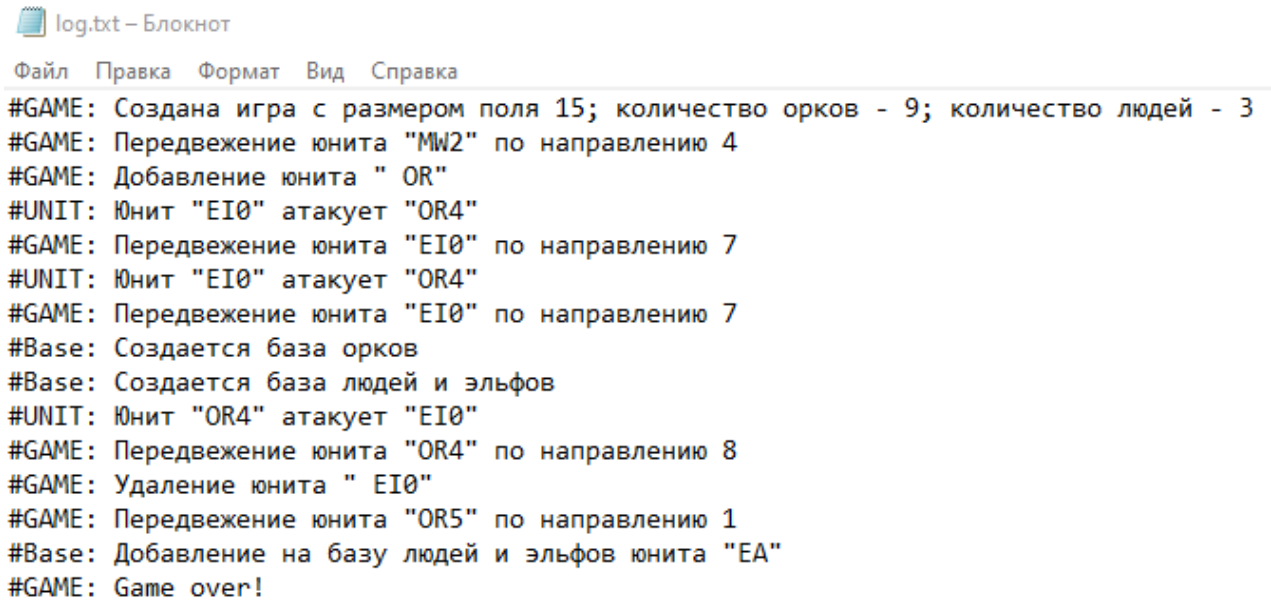
FormatterEmpty, наследующиеся от класса LogFormatter. Для вывода отформатированной строки также используется класс LoggerProxy, который реализует паттерн «Адаптер». Перед вызовом метода логирования класс LoggerProxy предварительно получает отформатированную строку от одного из вышеупомянутых классов (можно задать методом setFormat), а затем передает ее соответствующему логгеру.

Тестирование.

Пример вывода логирования в терминал (логи выводятся со знаком “#” в начале строки).



Пример вывода логирования в файл “log.txt”.



Выводы.

В ходе выполнения лабораторной работы были реализованы наборы классов для ведения логирования действий и состояний программы.