

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

ОТЧЕТ
по лабораторной работе №7
по дисциплине «Объектно-ориентированное программирование»
Тема: Написание исключений

Студентка гр. 8304

Птухов Д.А.

Преподаватель

Размочаева Н.В

Санкт-Петербург

2020

Цель работы.

Разработать и реализовать набор исключений. Основные требования к исключениям:

- Исключения покрывают как минимум все тривиальные случаи возникновения ошибки
- Все реализованные исключения обрабатываются в программе
- Исключения должны хранить подробную информацию об ошибке, а не только строку с сообщением об ошибке

Выполнение работы.

1) Был реализован класс LogInterface, являющийся интерфейсом для последующих классов. Он содержит методы по добавлению информации в лог о уничтожении передвижении и создании юнита.

2) Был реализован набор классов LogToFile и LogToTerminal предназначенных для записи данных в лог. Каждый из классов занимается записью в собственный выходной поток. Первый – в файл log.txt, второй – в терминал по запросу пользователя. Также добавлена возможность запуска программы без логирования.

3) Для реализация паттерна прокси был реализован класс LogProху, который принимает все запросы от класса Dialog и адресует их к нужным объектам.

4) Для реализации паттерна адаптер был создан класс LogAdapter. Предназначением которого является изменение исходного выходного потока в терминал на выходной поток в файл.

5) Перегружен оператор вывода в поток.

Примеры работы.

- 1) Лог программы после создания 2-х юнитов перемещения первого из них и его нападения на второго.

```
1) Archer was added to point (10; 10)
2) Assassin was added to point (11; 11)
3) Archer was moved from (10; 10) to (12; 12)
4) Archer attacked(40 damage) Assassin
But damage dodged
```

Рисунок 1 – Файловый лог программы

- 2) Аналогичные действия однако печать лога в терминал.

```
1) Warrior was added to point (10; 10)
2) Spearman was added to point (11; 11)
3) Warrior was moved from (10; 10) to (12; 12)
4) Warrior attacked(5 damage) Spearman
```

Рисунок 2 – Результат 2-го примера

- 3) Аналогичные действия однако режим без логирования.

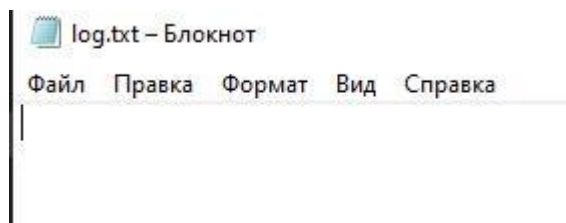


Рисунок 3 – Результат 3-го примера

Выводы.

Был получен опыт разработки набора классов, для ведения логирования действий и состояний программы