

**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: Логическое разделение классов**

Студентка гр. 8304

\_\_\_\_\_

Птухов Д.А.

Преподаватель

\_\_\_\_\_

Размочаева Н.В

Санкт-Петербург

2020

## **Цель работы.**

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

- Должен быть реализован функционал управления юнитами
- Должен быть реализован функционал управления базой

## **Выполнение работы.**

1) Был реализован класс Dialog ведущий диалог с пользователем. Пользователь имеет возможность добавить/создать в базе юнита, удалить его, переместить его а также посмотреть его текущие характеристики. Для удобства просмотра поля на каждой клетке соответствует 3 параметра первый – тип ландшафта, второй – тип нейтрального объекта(появляется при обнаружении), последний – тип юнита располагающегося на данной клетке. При создании данного класса был использован паттерн “Фасад”.

2) Был реализован набор классов addCheckTail, отвечающих за проверку данных поступающих от пользователя. Иерархия и интерфейс данных классов создана при помощи паттерна “Цепочка обязанностей”.

3) Аналогично пункту 2 были созданы наборы классов removeCheckTail и MUCTail (max unit check tail) осуществляющие проверку соответствующих их названию параметров. При их создании также был использован паттерн “Цепочка обязанностей”.

4) Также был использован паттерн “Посредник” для реализации класса Dialog, следовательно, посредником в данной реализации служит сам класс Dialog, а методы создания, удаления и т.д. получают корректные параметры или пользователь получает соответствующее ошибке сообщение.

## Примеры работы.

- 1) Окно, встречающее пользователя, а также под-окна всплывающие при выборе соответствующего пункта, представлены на рисунке 1.

```
Field size - 20x20

1) Work with base
2) Work with already created units
3) Show field
4) Quit

Enter command number: 1
1) Create base-unit
2) Use base-boost
3) Set max units count

Enter command number: 3
Enter count: 3

Data correct
Data set

1) Work with base
2) Work with already created units
3) Show field
4) Quit

Enter command number: 2
1) Create unit
2) Remove unit
3) Move unit
4) Unit characteristic
```

Рисунок 1 – Диалоговые окна

2) Пример создания юнита и получения его характеристик представлен на рисунке 2.

```
C:\Users\denis\source\repos\another try\Debug\another try.exe
Field size - 20x20

1) Work with base
2) Work with already created units
3) Show field
4) Quit

Enter command number: 2
1) Create unit
2) Remove unit
3) Move unit
4) Unit characteristic

Enter command number: 1

Enter coordinates: 10 10

Corrdinates correct
Units legend:
A - Archer
M - Mage
R - Rogue
S - Scout
W - Warrior
X - Assasin
Enter unit type(1 symbol): W

Unit added

1) Work with base
2) Work with already created units
3) Show field
4) Quit

Enter command number:
```

Рисунок 2 – Интерфейс создания юнита

### 3) Пример создания и последующего удаления юнита

```
1) Work with base
2) Work with already created units
3) Show field
4) Quit

Enter command number: 1
1) Create base-unit
2) Use base-boost
3) Set max units count

Enter command number: 1

Enter coordinates: 10 10

Corrdinates correct
Units legend:
A - Archer
M - Mage
R - Rogue
S - Scout
W - Warrior
X - Assasin
Enter unit type(1 symbol): A

I add unit!
Unit added

1) Work with base
2) Work with already created units
3) Show field
4) Quit

Enter command number: 2
1) Create unit
2) Remove unit
3) Move unit
4) Unit characteristic

Enter command number: 2
Enter coordinates: 10 10

Unit removed
I delete unit!
```

Рисунок 3 – Пример удаления и добавления юнита

## **Выводы.**

Были разработаны и реализованы наборы классов для взаимодействия пользователя с юнитами и базой.