

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Объектно-ориентированное программирование»**  
**Тема: Создание классов, конструкторов классов, методов классов,**  
**наследование**

Студентка гр. 8304

\_\_\_\_\_

Мельникова О.А.

Преподаватель

\_\_\_\_\_

Размочаева Н.В

Санкт-Петербург

2020

## Цель работы.

Разработать и реализовать набор классов: класс игрового поля, набор классов юнитов, удовлетворяющие требованиям, таким как, создание поля произвольного размера, контроль максимального количества объектов на поле. Возможность добавления, удаления объектов, копирования поля. Юниты должны иметь один общий интерфейс, имеют возможность перемещаться по карте.

## Выполнение работы.

1) Был реализован набор классов юнитов, создан класс Object — класс обозначающий любой объект на поле. От класса Object наследуется класс Unit. От Unit наследуется три класса: Infantry, Cavalery, Archery, от каждого из них соответственно наследуются еще по два класса knight и spearMan, lightRider и heavyRider, archer и crossBowMan. Наследование и содержание каждого класса (атрибуты и характеристики) отображено в таблице 1.

Таблица 1 — Структура наследования классов.

Object name, symb, health, cell					
Unit damage, movementRadius, attackRadius					
Infantry		Cavalery		Archery	
knight	spearMan	lightRider	heavyRider	archer	crossBowMan

2) Все юниты имеют один общий интерфейс — перемещение. Метод moveUnit описан в классе Field, принимает старые и новые координаты. С помощью метода canMove, который описан в Object и перегружен в Unit проверяется возможность переместиться на заданное количество клеток, затем объект перемещается в новую ячейку с помощью конструктора копирования.

3) Был создан класс игрового поля Field, а также класс ячейки поля Cell. Поле — это двумерный массив (прямоугольная сетка) объектов класса Cell. Cell содержит в себе координаты x,y объекта и innerObject (внутренний объект типа Object). Создание поля произвольного размера осуществляется конструктором

поля, который принимает ширину и высоту необходимой карты. Контроль максимального количества элементов на поле осуществляется проверками в методах добавления элемента: ячейка должна существовать, добавляемого объекта еще не должно быть на карте. Возможность добавления и удаления объектов реализована методами `addObject(Object& object, int x, int y)`, `addObject(Object& object, Cell& cell)`, `deleteObject(int x, int y)`, `deleteObject(Cell& cell)`, `deleteObject(Object& unit)`. Возможность копирования поля (включая объекты на нем) реализована конструкторами копирования и перегруженным оператором `=`. При приравнивании одного поля другому, все содержимое копируется в него.

4) Были выполнены дополнительные требования:

Созданы конструкторы копирования и перемещения для классов `Field`, `Cell`.

Все методы принимают параметры оптимальным образом — нет лишнего копирования т. к. объекты передаются по ссылкам и указателям.

Для создания юнитов используется паттерн «Абстрактная фабрика», реализация в методе `objectFabric`.

### Тестирование.

Создание поля:  
`Field field1(12, 12);`

```
22:48:58: Запускается /home/olga/QT/build-oop-Desktop
0                                     1
|-1-|-2-|-3-|-4-|-5-|-6-|-7-|-8-|-9-|-0-|-1-|-2-|
1  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
2  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
3  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
4  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
5  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
6  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
7  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
8  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
9  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
0  |  |  |  |  |  |  |  |  |  |  |  |  |  |
1  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
2  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
```

Добавление объектов:

```
field1.addObject(fabric.createArcher(), 3, 2);
field1.addObject(fabric.createCrossbowman(), 2, 2);
field1.addObject(fabric.createLightRider(), 2, 3);
field1.addObject(fabric.createHeavyRider(), 2, 4);
field1.addObject(fabric.createKnight(), 4, 4);
field1.addObject(fabric.createSpearman(), 12, 12);
```

[illegible]

Копирование поля:

```
Field field2 = field1;  
field2.print();
```

[illegible]

```
field2.moveUnit(2,4, 3,6);
```

0	1
-1-	-2-
1	
2	C A
3	L
4	K
5	
6	H
7	
8	
9	
0	
1	
1	
2	S

```
field2.deleteObject(3, 2);
```

[illegible]

```
Конструкторы:  
Field f(5,5);  
Field g = f;  
std::cout << g.H << ";" << g.W << std::endl;  
Cell r(6,6);  
Cell s(std::move(r));  
std::cout << s.x << ";" << s.y << std::endl;
```

5;5

6;6

### **Выводы.**

Был разработан и реализован набор классов: класс игрового поля, набор классов юнитов, удовлетворяющие требованиям, таким как, создание поля произвольного размера, контроль максимального количества объектов на поле. Возможность добавления, удаления объектов, копирования поля. Юниты имеют один общий интерфейс, есть возможность перемещаться по карте.