

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «ООП»

Тема: Создание классов, конструкторов классов, методов классов;
наследование

Студент(ка) гр. 0000

Ивченко А.А.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2020

Цель работы.

Разработать и реализовать набор классов:

- Класс игрового поля
- Набор классов юнитов

Игровое поле является контейнером для объектов представляющим прямоугольную сетку. Основные требования к классу игрового поля:

- Создание поля произвольного размера
- Контроль максимального количества объектов на поле
- Возможность добавления и удаления объектов на поле
- Возможность копирования поля (включая объекты на нем)
- Для хранения запрещается использовать контейнеры из stl

Юнит является объектов, размещаемым на поля боя. Один юнит представляет собой отряд. Основные требования к классам юнитов:

- Все юниты должны иметь как минимум один общий интерфейс
- Реализованы 3 типа юнитов (например, пехота, лучники, конница)
- Реализованы 2 вида юнитов для каждого типа(например, для пехоты могут быть созданы мечники и копейщики)
- Юниты имеют характеристики, отражающие их основные атрибуты, такие как здоровье, броня, атака.
- Юнит имеет возможность перемещаться по карте

Баллы за лаб. работу (* отмечает необязательные пункты)

Выполнение работы.

В ходе работы был разработан класс игрового поля Field, для которого были определены следующие методы: конструктор и деструктор класса, конструктор копирования и перемещения, перегрузка оператора присваивания, метод добавления, исключения и перемещения юнитов по игровому полю, возможность создания поля производного размера. Для хранения данных использовалось динамическое выделение памяти.

Для реализации юнитов был создан класс Unit, поля которого отражают основные атрибуты боевых отрядов (а именно armor, damage, speed). Методом

наследования классов были созданы разные типы отрядов, которые делятся на подотряды и имеют свои оригинальные характеристики, а также имеют один общий интерфейс, прописанный в классе Unit. Юниты могут перемещаться по карте.

Тестирование:

```
Field f(10, 5, 10);
Swordman swm;
Spearman spm;

f.AddObj(&swm, 1, 1);
f.AddObj(&spm, 2, 3);
f.Moving(&spm, 2, 3, 1, -1);
f.RemObj(&swm, 1, 1);
f.PrintField();

std::cout << std::endl;

Field newfield(0, 0, 0);

newfield = f;
newfield.PrintField();
```

```
Unit added
Unit added
Unit moved
Unit removed
0000000000
0000000000
0000000000
0020000000
0000000000
Field created
0000000000
0000000000
0000000000
0020000000
0000000000
Field deleted
Field deleted
```

Вывод.

В результате работы были получены знания по классовому наследованию, написанию конструкторов копирования и перемещения, перегрузке операторов. Были разработаны классы, необходимые по условию задания, а также выполнены основные требования к ним.