

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

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

Студент гр. 8382

Янкин Д.О.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Ознакомиться с классами, конструкторами и методами классов, наследованием.

Постановка задачи.

Разработать и реализовать набор классов:

- Класс игрового поля;
- Набор классов юнитов.

Игровое поле является контейнером для объектов, представляющих 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 объектом, размещаемым на поле боя. Один юнит представляет собой отряд. Основные требования к классам юнитов:

- Все юниты должны иметь как минимум один общий интерфейс;
- Реализованы 3 типа юнитов;
- Реализованы 2 вида юнитов для каждого типа;
- Юниты имеют характеристики, отражающие их основные атрибуты;
- Юнит имеет возможность перемещаться по карте.

Выполнение работы.

Созданы классы клетки поля `Cell` и поля `Field`, хранящего в себе массив клеток. `Field` предоставляет методы для работы с объектами, размещенными на поле. Смотреть директорию `/Field`.

Был реализован класс юнита, несколько групп-наследников, для каждой группы реализованы несколько конкретных юнитов. Группа и юниты

различаются модификаторами при расчете урона. Смотреть директорию /Unit: IUnit.h, Unit.h, Melee.h, Ranged.h, SiegeEngine.h.

Для класса поля реализованы конструкторы копирования и перемещения. Смотреть директорию /Field.

Для атрибутов юнитов созданы классы: здоровье, сила, броня, очки передвижения. Смотреть /Trivia/Attributes.h.

Для удобства создания юнитов используется фабричный метод. Смотреть /Unit/UnitFactory.h.

Для поля создан итератор. Смотреть Field/Field.h, реализация в Field/FieldIterator.cpp.

Выводы.

Были успешно реализованы требуемые классы, получен опыт работы с конструкторами, наследованием и виртуальными методами.