

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

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

Студент гр. 8381

Лисок М.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2020

## **Цель работы.**

Ознакомиться с основными принципами объектно-ориентированного программирования. Реализовать модель игрового поля с объектами на нем с помощью классов, их методов, а также наследования.

## **Задание.**

Разработать и реализовать набор классов:

- Класс игрового поля
- Набор классов юнитов

Игровое поле является контейнером для объектов, представляющим собой 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 `std`

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 (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

## **Выполнение работы.**

Написание работы производилось на базе операционной системы Windows 10 в среде разработки QtCreator с использованием фреймворка Qt. Сборка, отладка производились в QtCreator, запуск программы осуществлялся

через командную строку. Исходные коды файлов программы представлены в приложениях А-Л.

Для игрового поля был создан класс Field. Основные методы класса приведены в табл. 1.

Таблица 1 – Основные методы класса Field

Field(unsigned width, unsigned height, unsigned itemLimit);	Конструктор, устанавливающий высоту и ширину поля, а также лимит числа объектов
Field(Field &field);	Конструктор копирования поля
Field(Field &&field);	Конструктор перемещения поля
bool addItem(unsigned x, unsigned y, FieldItem *item);	Метод добавления объекта на поле по заданным координатам. Бросает исключение в случае ошибок.
bool deleteItem(unsigned x, unsigned y);	Метод удаления объекта на поле по координатам. Бросает исключение в случае ошибок.
bool deleteItem(FieldItem *item);	Метод поиска и удаления объекта на поле. Бросает исключение в случае ошибок.
FieldItem *getItem(unsigned x, unsigned y);	Метод получения объекта по заданным координатам. Бросает исключение в случае ошибок.
bool moveItem(FieldItem *item, int x, int y);	Метод поиска и перемещения объекта на поле. Бросает исключение в случае ошибок.

Для класса Field был создан итератор в виде отдельного класса FieldIterator, который позволяет последовательно получать доступ к размещенным на поле объектам.

Для всех юнитов был создан класс Unit, наследованный от интерфейса FieldItem. Для атрибутов юнитов создан класс Skills, три объекта которого хранятся в классе Unit. Основные атрибуты юнитов в рамках игры представлены в табл. 2.

Таблица 2 – Основные атрибуты юнитов

Тип/вид	attackSkills	securitySkills	characteristics
move	С и л а а т а к и п о д в и г а т е л ь н о й ф у н к ц и и в р а г а	З а щ и т а о т а т а к и п о д в и г а т е л ь н о й ф у н к ц и и	Х а р а к т е р и с т и к а д в и г а т е л ь н о й ф у н к ц и и ю н и т а
power	С и л а а т а к и п о м о щ н о с т и в р а г а	З а щ и т а о т а т а к и п о м о щ н о с т и	Х а р а к т е р и с т и к а м о щ н о с т и ю н и т а
spread	С и л а а т а к и п о з а р а ж е н и ю в р а г а	З а щ и т а о т а т а к и п о з а р а ж е н и ю	Х а р а к т е р и с т и к а з а р а ж е н н о с т и ю н и т а

В классе реализованы методы для установки и получения характеристик, получения информации в виде строки, имени, а также для перемещения.

Перемещение юнита осуществляется с помощью паттерна «посредник», реализованного классом MoveMediator. Класс передает классу Field запрос от юнита на передвижение, а в случае ошибки передает исключение, брошенное методом moveItem.

Различные типы и виды юнитов реализованы с применением паттерна абстрактной фабрики. Существует три вида конкретных фабрик, создающих юнитов:

- AttackDevice
- SupportDevice
- SimpleDevice

Все методы классов сохраняют инварианты этих классов. Так, для координат используется беззнаковый тип unsigned, а при некорректных входных параметрах (например, координата вне поля) методы бросают исключения.

### **Выводы.**

В ходе выполнения лабораторной работы была написана программа, в которой реализованы классы поля и различных юнитов, взаимодействие между ними. Был использован объектно-ориентированный стиль программирования,

были изучены и применены его основные положения, такие как наследование, инкапсуляция, инвариантность.