

**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**

Студент гр. 8304

\_\_\_\_\_

Мухин А. М.

Преподаватель

\_\_\_\_\_

Размочаева Н.

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

### **Описание функций и структур данных.**

Описание функций и структур данных были сгенерированы благодаря утилите Doxygen. Туда не вошло описание используемого шаблонного итератора, лишь потому, что в нём переопределены всего 3 оператора. А именно: `T& operator++()`, `bool operator!=(const Iterator& second_object)`, `T& operator*()`.

### **Тестирование.**

Тестирование реализовано с помощью библиотеки googletest. Оно также введено в систему сборки проекта, поэтому при любом изменении программы мы можем проверить не нарушились ли другие части программы. Также благодаря утилите valgrind для WSL, мы смогли проверить нашу программу на утечки памяти. В результате проверки утечек не обнаружено.

### **Выводы.**

В ходе данной лабораторной работы мы научились реализовывать паттерн абстрактный метод для создания юнитов разных типов, с помощью полиморфизма и наследования. Реализовали собственный шаблонный контейнер, написали конструкторы копирования, переноса, оператор присваивания копированием и шаблонный итератор для этого контейнера.