

**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 класса, методов класса,  
наследование»**

Студент гр. 8304

Щука А.А.

Преподаватель

Размочаева Н.В.

Санкт-Петербург  
2020

## **Цель работы.**

Научиться создавать классы и их конструкторы, реализовать методы классов и познакомиться с наследованием классов.

## **Постановка задачи.**

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
- *Создан итератор для поля*

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
- *Для создания юнитов используются паттерны “Абстрактный метод” / “Абстрактная фабрика”*
- *Для атрибутов юнитов созданы свои классы. Создавать их требуется, если это не противоречит логике.*

- Созданы конструкторы копирования и перемещения
- Все методы принимают параметры оптимальным образом (то есть, отсутствует лишнее копирование объектов)

### Ход выполнения.

*Примечание. Полная UML-диаграмма приведена в отдельном файле.*

- 1) Был создан базовый абстрактный класс Юнитов, три подкласса, различающихся по способу движения и для каждого подкласса по 2 подкласса, различающихся по количеству добываемого золота. Иерархия классов представлена на рис. 1.

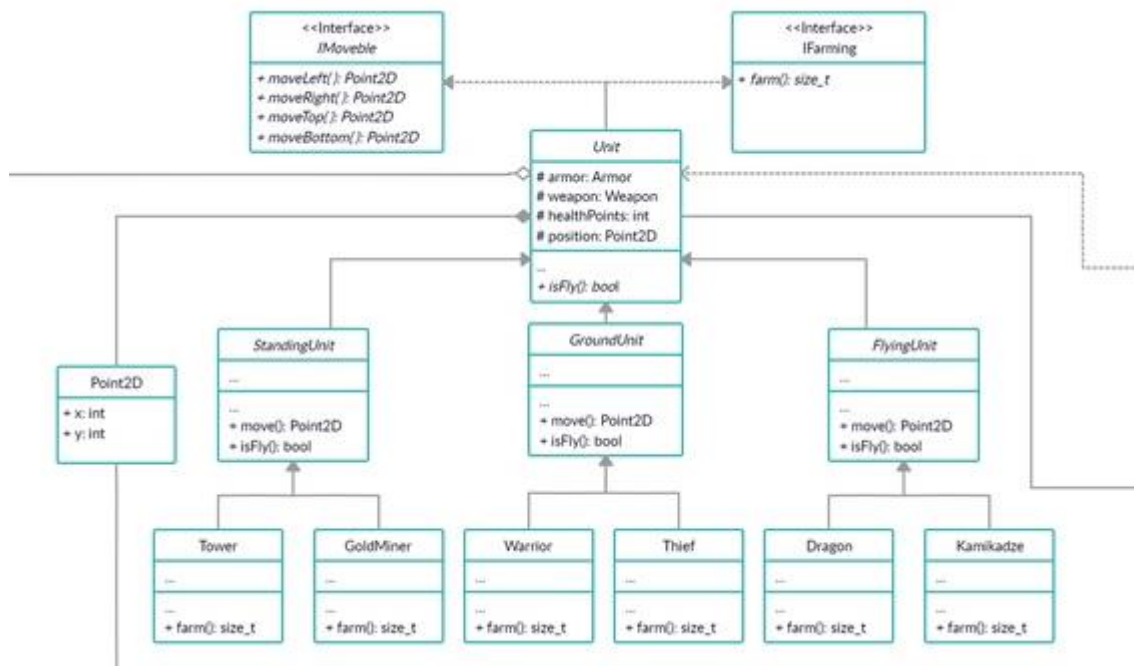


Рисунок 1 – Иерархия классов Юнит

- 2) Были созданы иерархии классов атрибутов, которые включает в себя юнит. Для клонирования был использован паттерн «Прототип». Типы, зависимости, поля и основные методы классов приведены на рис. 2.

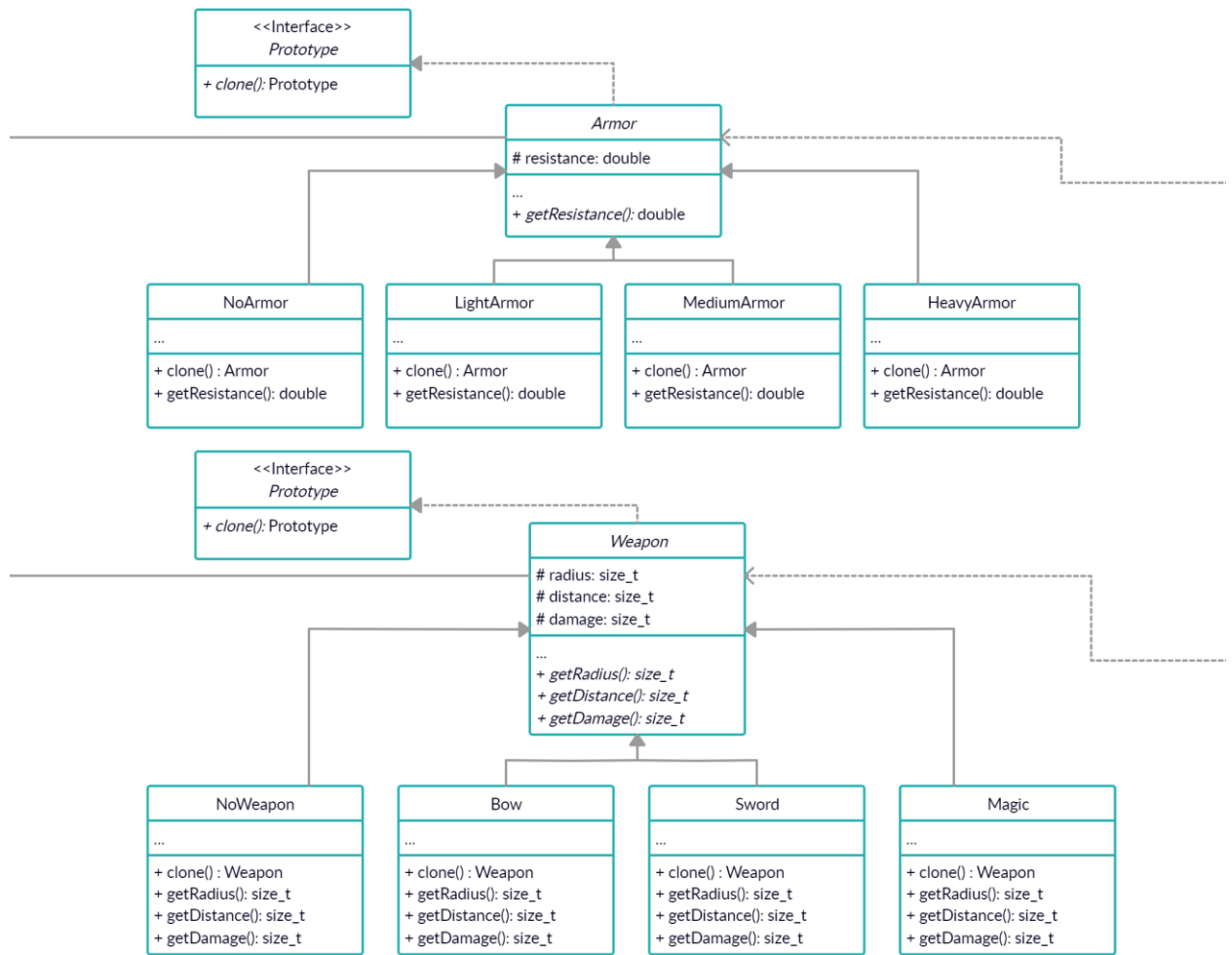


Рисунок 2 – Иерархия классов атрибутов

- 3) Для создания юнитов и атрибутов были использованы паттерны «фабричный метод» и «строитель». Зависимости классов, основные поля и методы приведены на рис. 3.

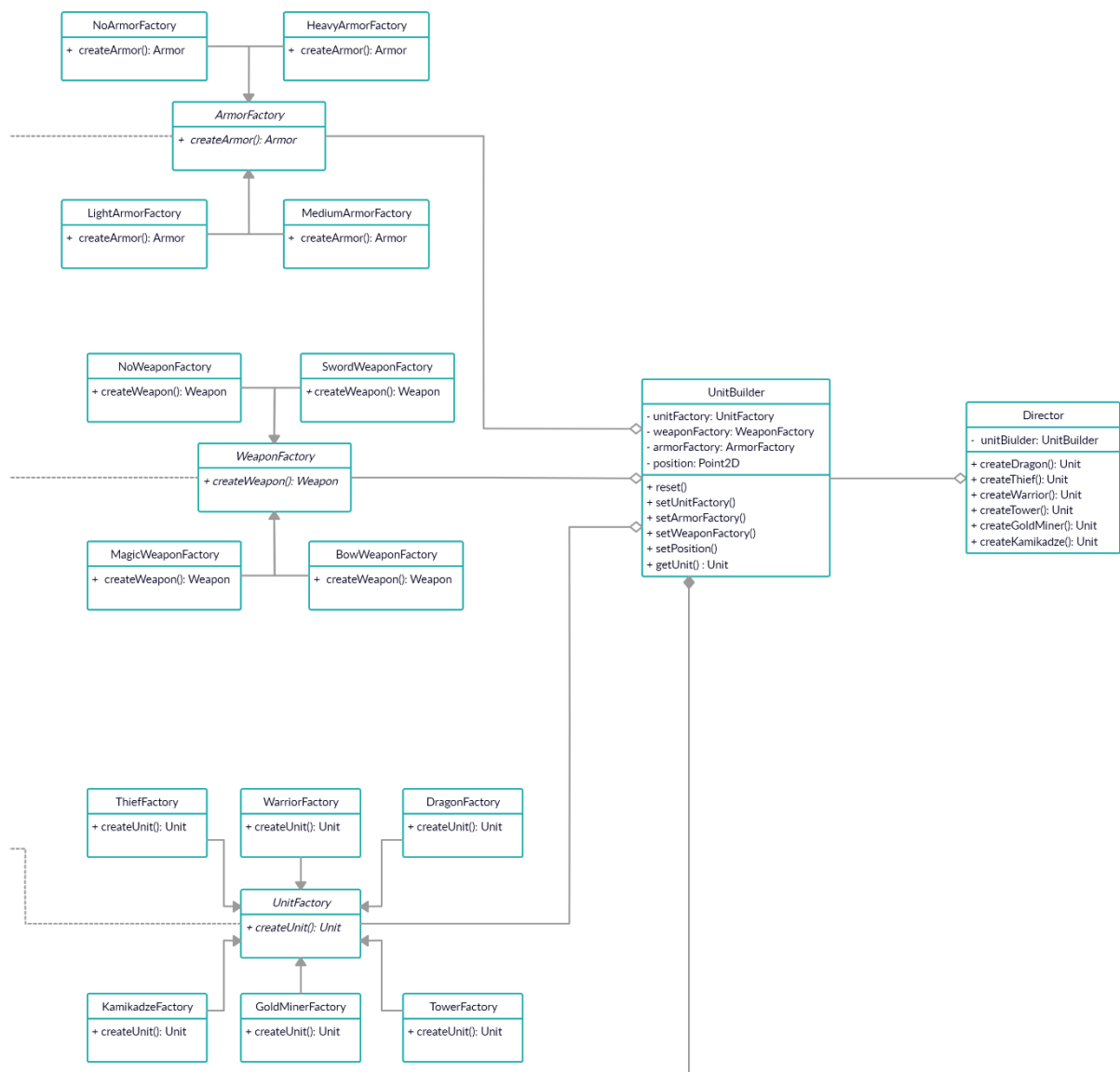


Рисунок 3 – Иерархия порождающих классов

- 4) Были созданы классы ячейки поля, игрового поля и итератора. Контейнеры из STL использованы не были. В конструкторе поля можно задать произвольный размер. Поле имеет счетчик юнитов, юниты могут быть добавлены или удалены с поля. Для поля и всех зависимых классов были определены конструкторы копирования и перемещения. Основные поля и методы, а также зависимости классов представлены на рис. 4.

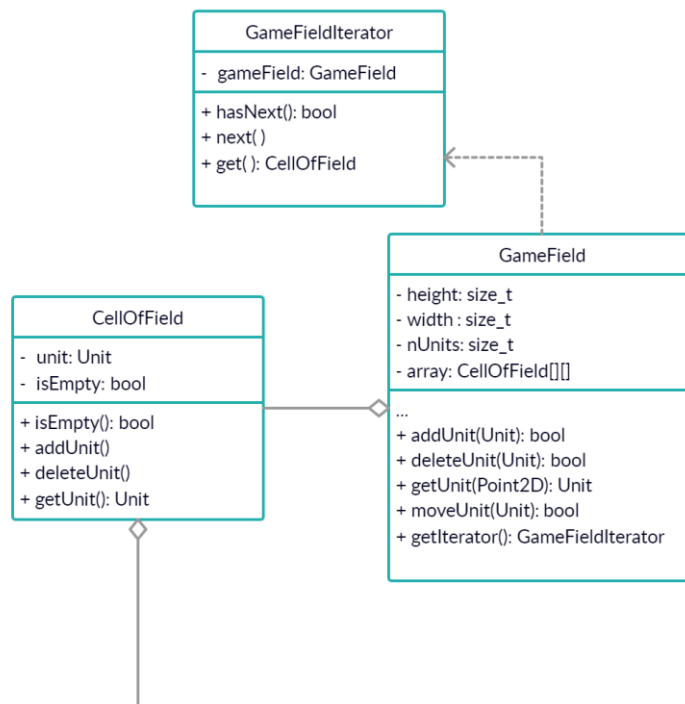


Рисунок 4 – Иерархия классов поля

- 5) При разработке методов было соблюдено требование: все методы принимают аргументы оптимальным образом.

### Выводы.

В ходе выполнения работы были созданы классы поля и юнитов и их конструкторы, реализовано наследование классов юнитов. Были выполнены все требования, в том числе и необязательные.

*Примечание.* Дополнительно использованные паттерны: «Прототип», «Строитель». Весь код написан с использованием «умных» указателей.