**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №5**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: Добавления врагов**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9382 |  | Демин В.В. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2020

## Цель работы.

## Создать шаблонный класс врага.

## Задание.

Создать шаблонный класс врага. Параметр шаблона должен определять поведение врага *(параметров шаблона может быть несколько, например отдельный параметр для политики передвижения и для политики атаки)*. Класс врага должен препятствовать игроку. Класс игрока должен иметь возможность взаимодействовать с врагом и наоборот.

**Обязательные требования:**

* Создан шаблонный класс врага
* Создано не менее 3 типа поведения врагов
* Взаимодействие происходит через перегруженный оператор

**Дополнительные требования:**

* Передача хода между игроков и врагами происходит с использованием паттерна **Состояния** в классе игры

## Выполнение работы.

Был создан шаблонный класс Enemy. Который в качестве параметров получал speed, size, damage. Было создано три объекта этого класса с разными характеристиками. Были определены кординаты врагов на поле. В классе игрока был прописан перегруженный оператор для взаимодействия с врагом.

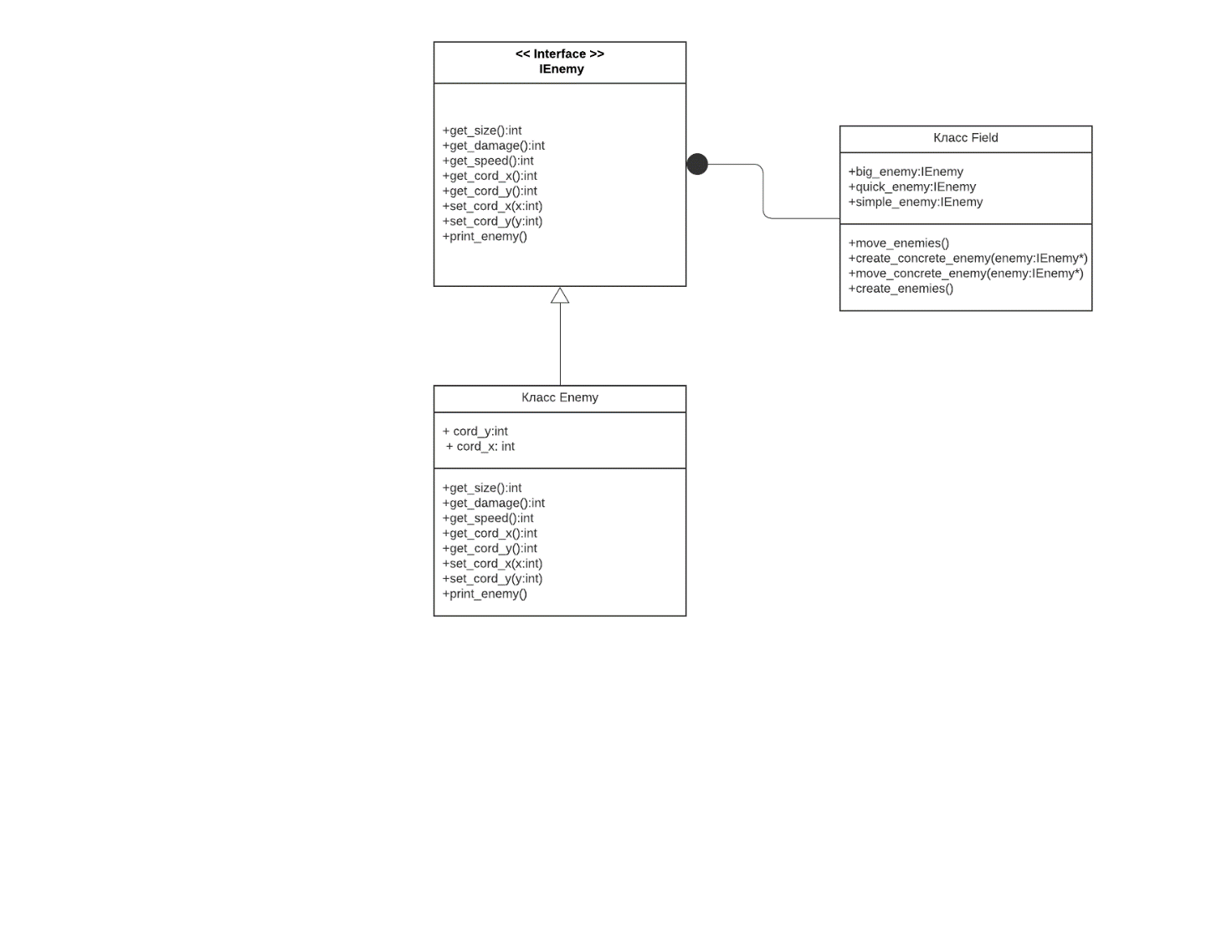


Рисунок 1. Uml диаграмма

## Выводы.

В ходе выполнения лабораторной работы был создан шаблонный класс врагов. Изучены методы работы с шаблонными классами.

# Приложение А Исходный код программы

Название файла: Enemy.h

//

// Created by vikto on 23.11.2020.

//

#ifndef MYGAME\_ENEMY\_H

#define MYGAME\_ENEMY\_H

class IEnemy {

public:

virtual ~IEnemy() {};

virtual int get\_size() const = 0;

virtual int get\_damage() const = 0;

virtual void set\_cord\_x(int x) = 0;

virtual void set\_cord\_y(int y) = 0;

virtual int get\_cord\_x() const = 0;

virtual int get\_cord\_y() const = 0;

virtual int get\_speed() const = 0;

virtual void print\_enemy() = 0;

};

template<int damage, int size, int speed>

class Enemy : public IEnemy {

int cord\_x = 0;

int cord\_y = 0;

public:

~Enemy() override {

};

int get\_size() const override {

return size;

}

int get\_damage() const override {

return damage;

}

int get\_speed() const override {

return speed;

}

int get\_cord\_x() const override {

return cord\_x;

}

int get\_cord\_y() const override {

return cord\_y;

}

void set\_cord\_x(int x) override {

cord\_x = x;

}

void set\_cord\_y(int y) override {

cord\_y = y;

}

void print\_enemy() override {

if (size > 1) {

std::cout << '#';

} else if (speed > 2) {

std::cout << '.';

} else {

std::cout << '-';

}

}

};

#endif //MYGAME\_ENEMY\_H