

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «ООП»**  
**Тема: Создание классов, конструкторов классов, методов классов;**  
**наследование**

Студент гр. 8383

\_\_\_\_\_

Шишкин И.В.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2020

### **Цель работы.**

Создать основные классы, методы в них для придуманной студентом игры. Так же ознакомиться с различными конструкторами классов и реализовать их.

### **Постановка задачи.**

Разработать и реализовать набор классов. Выполнить основные требования к классу игрового поля и классу юнитов.

### **Описание игры.**

В игре присутствуют 3 типа юнитов: люди, орки и эльфы. У типа "люди" есть 2 подвида: копьеносец и маг, у "эльфов" - лучник и невидимка, у "орков" - разбойники и пожиратели. У каждого типа и у каждого вида свои особенности, которые будут указаны далее. В игре участвуют 2 игрока: первый игрок за людей и эльфов, второй - за орков. На данном этапе разработки пользователь может выбрать количество созданные юнитов, размерность поля, может двигать юнитов по полю и удалять их.

### **Описание класса Mens.**

Класс людей. Задается буст (boost). Урон и броня умножаются на это значение, а здоровье делится на него. У класса людей есть 2 подкласса: копьеносец и маг. Способность копьеносца: если за врагом сразу же стоит еще один враг, то урон наносится обоим. Способность мага: может нанести урон, поделенный на 3, врагу на любом расстоянии.

### **Описание класса Elfs.**

Класс эльфов. Задается буст (boost). Урон умножается на (boost + 1), броню делит на boost, здоровье остается неизменным. У класса эльфов есть 2 подкласса: лучник и невидимка. Способность лучника: наносит урон врагу, находящегося максимум через 2 клетки от лучника. Способность невидимки:

становится невидимым на 3 хода (через него нельзя шагать и на него нельзя нападать).

### Описание класса Orcs.

Класс орков. Задается буст (boost). Урон и здоровье умножаются на это значение, броня делится на (boost + 1). У класса орков есть 2 подкласса: разбойники и пожиратели. Способность разбойников: если разбойник окружен (в упор к нему стоят минимум 2 врага), то он наносит урон обоим. Способность пожирателей: с вероятностью x% при атаке урон умножается на 2.

### Спецификация программы.

Программа написана на языке C++. Входные данные подаются из терминала. Результат программы выводится на экран.

### Тестирование.

#### 1) Движение определенного персонажа вверх.

```
Вам предложено ввести количество орков, эльфов и людей.
Так как в игре 2 игрока: один за орков, а другой за людей и эльфов, то
количество людей и эльфов должно соответствовать количеству орков.
(Первый игрок за людей и эльфов, второй - за орков. 4 < Кол-во орков < 10)

Введите количество людей:
3
Введите количество эльфов:
4
Создается поле размерностью GxG. Введите G:
7
```

```
EA1 --- --- --- --- OR1
--- EI1 --- --- --- OR2 ---
EA2 --- --- --- --- OD1
--- EI2 --- --- --- OD2 ---
MS1 --- --- --- --- OD3
--- MW1 --- --- --- OR3 ---
MW2 --- --- --- --- OD4

Подсказки:
OD - Orcs Devourers - орки пожиратели; OR - Orcs Robbers - орки разбойники
MS - Mens Spearman - люди копьеносцы; MW - Mens Wizard - люди маги
EA - Elfs Archer - эльф лучник; EI - Elfs Invisible - эльф невидимка
Введите "exit" для выхода, "add" - для добавления персонажа, "del" - для удаления

1-й игрок, ваш ход:
Введите персонажа из эльфов или людей:
EI1
Введите направление:
Возможные варианты хода:
1 - вверх
2 - вправо
3 - влево
4 - вниз
5 - не двигаться
1
```

Результат действия:

EA1	EI1	---	---	---	---	OR1
---	---	---	---	---	---	OR2
EA2	---	---	---	---	---	OD1
---	EI2	---	---	---	---	OD2
MS1	---	---	---	---	---	OD3
---	MW1	---	---	---	---	OR3
MW2	---	---	---	---	---	OD4

2) Удаление персонажа орков.

EA1	EI1	---	---	---	---	OR1
---	---	---	---	---	---	OR2
EA2	---	---	---	---	---	OD1
---	EI2	---	---	---	---	OD2
MS1	---	---	---	---	---	OD3
---	MW1	---	---	---	---	OR3
MW2	---	---	---	---	---	OD4

Подсказки:  
OD - Orcs Devourers - орки пожиратели; OR - Orcs Robbers - орки разбойники  
MS - Mens Spearman - люди копьеносцы; MW - Mens Wizard - люди маги  
EA - Elfs Archer - эльф лучник; EI - Elfs Invisible - эльф невидимка  
Введите "exit" для выхода, "add" - для добавления персонажа, "del" - для удаления  
2-й игрок, ваш ход:  
Введите персонажа из орков:  
del  
Введите обозначение персонажа, которого вы хотите удалить:  
OD1

Результат:

EA1	EI1	---	---	---	---	OR1
---	---	---	---	---	---	OR2
EA2	---	---	---	---	---	---
---	EI2	---	---	---	---	OD2
MS1	---	---	---	---	---	OD3
---	MW1	---	---	---	---	OR3
MW2	---	---	---	---	---	OD4

3) Движение соседнего персонажа из п. 1 вниз.

EA1	EI1	---	---	---	---	OR1
---	---	---	---	---	---	OR2
EA2	---	---	---	---	---	---
---	EI2	---	---	---	---	OD2
MS1	---	---	---	---	---	OD3
---	MW1	---	---	---	---	OR3
MW2	---	---	---	---	---	OD4

Подсказки:  
OD - Orcs Devourers - орки пожиратели; OR - Orcs Robbers - орки разбойники  
MS - Mens Spearman - люди копьеносцы; MW - Mens Wizard - люди маги  
EA - Elfs Archer - эльф лучник; EI - Elfs Invisible - эльф невидимка  
Введите "exit" для выхода, "add" - для добавления персонажа, "del" - для удаления  
1-й игрок, ваш ход:  
Введите персонажа из эльфов или людей:  
EA1  
Введите направление:  
Возможные варианты хода:  
4 - вниз  
5 - не двигаться  
4

Результат:

---	EI1	---	---	---	---	OR1
EA1	---	---	---	---	---	OR2
EA2	---	---	---	---	---	---
---	EI2	---	---	---	---	OD2
MS1	---	---	---	---	---	OD3
---	MW1	---	---	---	---	OR3
MW2	---	---	---	---	---	OD4

**Вывод.**

В ходе выполнения лабораторной работы была создана "почва" для игры на языке программирования C++.

## Приложение А

### Содержимое файла Source.cpp

```
#include <iostream>
#include <iomanip>
#include "Warrior.h"
#include "Orcs.h"
#include "Men.h"
#include "Elfs.h"
#include "Field.h"
#include "Iterator.h"

using namespace std;

int scan(int& numOfOrcs, int& numOfMens, int& numOfElfs, int& G);

int main() {
    setlocale(LC_ALL, "RUS");

    int G;
    string st;

    int numOfOrcs;
    int numOfMens;
    int numOfElfs;

    if (scan(numOfOrcs, numOfMens, numOfElfs, G) == 0) {
        cout << "Ошибка!\n";
        system("pause");
        return 0;
    }
    cout << numOfOrcs << endl << numOfMens << endl << numOfElfs << endl << G << endl;

    Field f1(G);
    f1.init(numOfOrcs, numOfElfs, numOfMens);

    int k = 0;
    int direction;
    do {
        system("cls");
        st.clear();
        f1.print();
        int forAddAndDel = 0;
        if (k % 2 == 0) {
            cout << "1-й игрок, ваш ход:\n";
            cout << "Введите персонажа из эльфов или людей:\n";
            cin >> st;
            if (st == "exit") break;
            else if (st == "add") {
                forAddAndDel = 1;
                cout << "Введите обозначение персонажа, которого вы хотите
добавить:\n";

                string character;
                cin >> character;
                if (character != "OD" && character != "OR" && character != "MS"
&& character != "MW" && character != "EA" && character != "EI") {
                    cout << "Неправильное обозначение персонажа. Вы
пропускаете ход!\n";

                    system("pause");
                }
                else {
                    f1.add(character);
                }
            }
        }
        k++;
    } while (true);
}
```

```

    }
    else if (st == "del") {
        forAddAndDel = 1;
        cout << "Введите обозначение персонажа, которого вы хотите
удалить:\n";

        string character;
        cin >> character;
        f1.del(character);
    }
    else
    while (st[0] == '0') {
        cout << "Неверно! Вы играете за эльфов и людей!\n";
        cin >> st;
    }
}
else {
    cout << "2-й игрок, ваш ход:\n";
    cout << "Введите персонажа из орков:\n";
    cin >> st;
    if (st == "exit") break;
    else if (st == "add") {
        forAddAndDel = 1;
    }
    else if (st == "del") {
        forAddAndDel = 1;
        cout << "Введите обозначение персонажа, которого вы хотите
удалить:\n";

        string character;
        cin >> character;
        f1.del(character);
    }
    else
    while (st[0] == 'M' || st[0] == 'E') {
        cout << "Неверно! Вы играете за орков!\n";
        cin >> st;
    }
}
if (forAddAndDel == 0) {
    getchar();
    cout << "Введите направление:\n";
    string opportunity = f1.check(st);
    cin >> direction;

    char tmp = direction + '0';
    int checker = 0;
    for (int i = 0; i < opportunity.length(); i++) {
        if (opportunity[i] == tmp) checker = 1;
    }

    if (checker == 1) f1.moving(st, direction);
    else {
        cout << "Такого направления нет! Вы пропускаете ход!\n";
        system("pause");
    }
}
k++;
} while (st != "exit");

system("pause");
return 0;
}

int scan(int& numOfOrcs, int& numOfMens, int& numOfElfs, int& G) {

```

```

cout << "Вам предложено ввести количество орков, эльфов и людей.\n";
cout << "Так как в игре 2 игрока: один за орков, а другой за людей и эльфов, то\n";
cout << "количество людей и эльфов должно соответствовать количеству орков.\n";
cout << "(Первый игрок за людей и эльфов, второй - за орков. 4 < Кол-во орков <
10)\n\n";

cout << "Введите количество людей:\n";
cin >> numOfMens;
if (cin.fail() || numOfMens <= 0) {
    return 0;
}

cout << "Введите количество эльфов:\n";
cin >> numOfElfs;
if (cin.fail() || numOfElfs <= 0) {
    return 0;
}

numOfOrcs = numOfElfs + numOfMens;

if (numOfOrcs <= 4 || numOfOrcs >= 10) return 0;

cout << "Создается поле размерностью GxG. Введите G:\n";
cin >> G;

if (cin.fail()) {
    return 0;
}
while (G <= numOfOrcs - 1) {
    cout << "G введено неверно! Оно должно быть больше " << numOfOrcs - 1 << endl;
    cin >> G;
}

return 1;
}

```



## Приложение Б

### Содержимое файла Field.h

```
#pragma once

#include "Warrior.h"
#include "Orcs.h"
#include "Men.h"
#include "Elfs.h"

using namespace std;

class Field {
public:
    Field();
    Field(int a);
    ~Field();
    Field(const Field& f);
    Field(Field&& f);
    void mas();
    void print();
    void init(int numOfOrcs, int numOfElfs, int numOfMens);
    void moving(string character, int direction);
    string check(string character);
    void add(string character);
    void del(string character);

private:
    int size, x0, y0;
    Warrior** cell;
};
```

## Приложение В

### Содержимое файла Field.cpp

```
#include "Field.h"
#include <iostream>
#include <time.h>

using namespace std;

Field::Field() {
    size = 3;
    x0 = 2; y0 = 2;
}

Field::Field(int a) {
    size = a;
    x0 = y0 = size - 1;
}

Field::~Field() {
    for (int i = 0; i < size; i++) {
        delete[] cell[i];
    }
}

Field::Field(const Field& f) {           //конструктор копирования
    this->size = f.size;
    this->x0 = f.x0;
    this->y0 = f.y0;
    this->mas();
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            cell[i][j] = f.cell[i][j];
        }
    }
}

Field::Field(Field&& f) : size(f.size), cell(f.cell){           //конструктор
перемещения
    cell = nullptr;
}

void Field::mas() {
    cell = new Warrior * [size];
    for (int i = 0; i < size; i++) {
        cell[i] = new Warrior[size];
    }
}

void Field::print() {
    for (int i = 0; i < (size + 1) * 4 - 2; i++) cout << "_";
    cout << endl;

    for (int i = 0; i < size; i++) {
        cout << "|";
        for (int j = 0; j < size; j++) {
            cout << " " << cell[i][j].GetValue();
        }
        cout << "| " << endl;
    }

    for (int i = 0; i < (size + 1) * 4 - 2; i++) cout << "_";
    cout << endl << endl << endl << endl;
```

```

        cout << "Подсказки:\n";
        cout << "OD - Orcs Devourers - орки пожиратели; OR - Orcs Robbers - орки
разбойники\n";
        cout << "MS - Mens Spearman - люди копьеносцы; MW - Mens Wizard - люди маги\n";
        cout << "EA - Elfs Archer - эльф лучник; EI - Elfs Invisible - эльф невидимка\n";
        cout << "Введите \"exit\" для выхода, \"add\" - для добавления персонажа, \"del\" -
для удаления персонажа\n";
        cout << endl;
    }

    void Field::init(int numOfOrcs, int numOfElfs, int numOfMens) {
        mas();
        srand(time(0));

        Devourers dev;
        Robbers rob;

        Spearman sp;
        Wizard wiz;

        Archer ar;
        Invisible inv;

        int half = size / 2;
        string d = dev.GetValue(); // 0
        string r = rob.GetValue(); // 1
        string s = sp.GetValue(); // 2
        string w = wiz.GetValue(); // 3
        string a = ar.GetValue(); // 4
        string invi = inv.GetValue(); // 5

        int num = 0;
        for (int i = 0; i < numOfElfs; i++) {
            num = 4 + rand() % 2;

            if (i % 2 == 0) {
                if (num == 4) {
                    ar.SetCounter(ar.GetCounter() + 1);
                    char sym = ar.GetCounter() + '0';
                    cell[i][0].SetValue(a + sym);
                }
                else if (num == 5) {
                    inv.SetCounter(inv.GetCounter() + 1);
                    char sym = inv.GetCounter() + '0';
                    cell[i][0].SetValue(invi + sym);
                }
                cell[i][0].SetPlaceX(i);
                cell[i][0].SetPlaceY(0);
            }
            else {
                if (num == 4) {
                    ar.SetCounter(ar.GetCounter() + 1);
                    char sym = ar.GetCounter() + '0';
                    cell[i][1].SetValue(a + sym);
                }
                else if (num == 5) {
                    inv.SetCounter(inv.GetCounter() + 1);
                    char sym = inv.GetCounter() + '0';
                    cell[i][1].SetValue(invi + sym);
                }
                cell[i][1].SetPlaceX(i);
                cell[i][1].SetPlaceY(1);
            }
        }
    }

```

```

num = rand() % 2;
if (i % 2 == 0) {
    if (num == 0) {
        dev.SetCounter(dev.GetCounter() + 1);
        char sym = dev.GetCounter() + '0';
        cell[i][size - 1].SetValue(d + sym);
    }
    else if (num == 1) {
        rob.SetCounter(rob.GetCounter() + 1);
        char sym = rob.GetCounter() + '0';
        cell[i][size - 1].SetValue(r + sym);
    }
    cell[i][size - 1].SetPlaceX(i);
    cell[i][size - 1].SetPlaceY(size - 1);
}
else {
    if (num == 0) {
        dev.SetCounter(dev.GetCounter() + 1);
        char sym = dev.GetCounter() + '0';
        cell[i][size - 2].SetValue(d + sym);
    }
    else if (num == 1) {
        rob.SetCounter(rob.GetCounter() + 1);
        char sym = rob.GetCounter() + '0';
        cell[i][size - 2].SetValue(r + sym);
    }
    cell[i][size - 2].SetPlaceX(i);
    cell[i][size - 2].SetPlaceY(size - 2);
}
}

for (int i = numOfElfs; i < numOfOrcs; i++) {
    num = 2 + rand() % 2;

    if (i % 2 == 0) {
        if (num == 2) {
            sp.SetCounter(sp.GetCounter() + 1);
            char sym = sp.GetCounter() + '0';
            cell[i][0].SetValue(s + sym);
        }
        else if (num == 3) {
            wiz.SetCounter(wiz.GetCounter() + 1);
            char sym = wiz.GetCounter() + '0';
            cell[i][0].SetValue(w + sym);
        }
        cell[i][0].SetPlaceX(i);
        cell[i][0].SetPlaceY(0);
    }
    else {
        if (num == 2) {
            sp.SetCounter(sp.GetCounter() + 1);
            char sym = sp.GetCounter() + '0';
            cell[i][1].SetValue(s + sym);
        }
        else if (num == 3) {
            wiz.SetCounter(wiz.GetCounter() + 1);
            char sym = wiz.GetCounter() + '0';
            cell[i][1].SetValue(w + sym);
        }
        cell[i][1].SetPlaceX(i);
        cell[i][1].SetPlaceY(1);
    }
}

num = rand() % 2;

```

```

        if (i % 2 == 0) {
            if (num == 0) {
                dev.SetCounter(dev.GetCounter() + 1);
                char sym = dev.GetCounter() + '0';
                cell[i][size - 1].SetValue(d + sym);
            }
            else if (num == 1) {
                rob.SetCounter(rob.GetCounter() + 1);
                char sym = rob.GetCounter() + '0';
                cell[i][size - 1].SetValue(r + sym);
            }
            cell[i][size - 1].SetPlaceX(i);
            cell[i][size - 1].SetPlaceY(size - 1);
        }
        else {
            if (num == 0) {
                dev.SetCounter(dev.GetCounter() + 1);
                char sym = dev.GetCounter() + '0';
                cell[i][size - 2].SetValue(d + sym);
            }
            else if (num == 1) {
                rob.SetCounter(rob.GetCounter() + 1);
                char sym = rob.GetCounter() + '0';
                cell[i][size - 2].SetValue(r + sym);
            }
            cell[i][size - 2].SetPlaceX(i);
            cell[i][size - 2].SetPlaceY(size - 2);
        }
    }

}

void Field::moving(string character, int direction) {
    int coordx;
    int coordy;
    // 1 - up, 2 - right, 3 - left, 4 - down, 5 - stay
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (cell[i][j].GetValue() == character) {
                //coordx = cell[i][j].GetPlaceX();
                //coordy = cell[i][j].GetPlaceY();
                coordx = i;
                coordy = j;
                break;
            }
        }
    }

    Warrior tmp;
    switch (direction) {
        case 1:
            cell[coordx - 1][coordy].SetValue(cell[coordx][coordy].GetValue());
            cell[coordx][coordy].SetValue("---");
            cell[coordx - 1][coordy].SetPlaceX(coordx-1);
            cell[coordx][coordy].SetPlaceX(coordx);
            cout << "x - " << coordx - 1 << ", y - " << coordy << endl;
            break;

        case 2:
            cell[coordx][coordy + 1].SetValue(cell[coordx][coordy].GetValue());
            cell[coordx][coordy].SetValue("---");
            cell[coordx][coordy + 1].SetPlaceY(coordy + 1);
            cell[coordx][coordy].SetPlaceY(coordy);
    }
}

```

```

        break;

    case 3:
        cell[coordx][coordy - 1].SetValue(cell[coordx][coordy].GetValue());
        cell[coordx][coordy].SetValue("---");
        cell[coordx][coordy - 1].SetPlaceY(coordy - 1);
        cell[coordx][coordy].SetPlaceY(coordy);
        break;

    case 4:
        cell[coordx + 1][coordy].SetValue(cell[coordx][coordy].GetValue());
        cell[coordx][coordy].SetValue("---");
        cell[coordx + 1][coordy].SetPlaceX(coordx + 1);
        cell[coordx][coordy].SetPlaceX(coordx);
        break;
    }
}

string Field::check(string character) {
    int coordx = 0;
    int coordy = 0;
    // 1 - up, 2 - right, 3 - left, 4 - down, 5 - stay
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (cell[i][j].GetValue() == character) {
                //coordx = cell[i][j].GetPlaceX();
                //coordy = cell[i][j].GetPlaceY();
                coordx = i;
                coordy = j;
                break;
            }
        }
    }

    string opportunity = "5";
    string tmp = "---";

    cout << "Возможные варианты хода:\n";
    if (coordx != 0) {
        if (cell[coordx - 1][coordy].GetValue() == tmp) {
            opportunity += '1';
            cout << "1 - вверх\n";
        }
    }
    if (coordy != size - 1) {
        if (cell[coordx][coordy + 1].GetValue() == tmp) {
            opportunity += '2';
            cout << "2 - вправо\n";
        }
    }
    if (coordy != 0) {
        if (cell[coordx][coordy - 1].GetValue() == tmp) {
            opportunity += '3';
            cout << "3 - влево\n";
        }
    }
    if (coordx != size - 1) {
        if (cell[coordx + 1][coordy].GetValue() == tmp) {
            opportunity += '4';
            cout << "4 - вниз\n";
        }
    }

    cout << "5 - не двигаться\n";
}

```

```

        return opportunity;
    }

void Field::add(string character) {
    int coordx = 0;
    int coordy = 0;

    cout << "Введите координаты (сначала x, затем y через enter):\n";
    cin >> coordx;
    if (cin.fail() || coordx > size - 1 || coordx < 0) return;
    cin >> coordy;
    if (cin.fail() || coordy > size - 1 || coordy < 0) return;
    if (cell[coordx][coordy].GetValue() != "---") return;

    /*Warrior w;
    //"OD" "OR" "MS" "MW" "EA" "EI"
    if (character == "OD") {
        w = Warrior::createWarrior(Devourers_ID);
    }
    else if (character == "OR") {

    }
    else if (character == "MS") {

    }
    else if (character == "MW") {

    }
    else if (character == "EA") {

    }
    else if (character == "EI") {

    }

    string ww = w.GetValue();
    w.SetCounter(w.GetCounter() + 1);
    char sym = w.GetCounter() + '0';
    cell[coordx][coordy].SetValue(ww + sym);*/
}

void Field::del(string character) {
    int coordx = 0;
    int coordy = 0;
    int check = 0;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (cell[i][j].GetValue() == character) {
                coordx = i;
                coordy = j;
                check = 1;
                break;
            }
        }
    }

    if (check == 0) {
        cout << "Такого персонажа на поле нет!\n";
        system("pause");
        return;
    }
}

```

```
    cell[coordx][coordy].SetValue("---");  
}
```



## Приложение Г

### Содержимое файла Warrior.h

```
#pragma once

#include <iostream>

enum Warrior_ID { Archer_ID = 0, Invisible_ID, Spearman_ID, Wizard_ID, Devourers_ID,
Robbers_ID };

class Warrior {
public:
    Warrior();
    ~Warrior();
    static Warrior createWarrior(Warrior_ID id); //фабричный метод
    void SetPlaceX(int placex);
    int GetPlaceX(void);
    void SetPlaceY(int placey);
    int GetPlaceY(void);
    void SetValue(std::string value);
    std::string GetValue(void);
    void SetHealth(int health);
    int GetHealth();
    void SetDamage(int damage);
    int GetDamage();
    void SetArmor(int armor);
    int GetArmor();

private:
    int placex;
    int placey;
    std::string value;

    /*Health health;
    Damage damage;
    Armor armor;*/
    int health;
    int damage;
    int armor;
};
```

## Приложение Д

### Содержимое файла Warrior.cpp

```
#include "Warrior.h"
#include "Elfs.h"
#include "Men.h"
#include "Orcs.h"

Warrior::Warrior() {
    this->placex = 0;
    this->placey = 0;
    this->health = 50;
    this->damage = 5;
    this->armor = 25;
    this->value = "---";
}

Warrior::~Warrior() {
}

Warrior Warrior::createWarrior(Warrior_ID id) {
    Warrior p;

    switch (id) {
        case Archer_ID:
            p = Archer();
            break;

        case Invisible_ID:
            p = Invisible();
            break;

        case Spearman_ID:
            p = Spearman();
            break;

        case Wizard_ID:
            p = Wizard();
            break;

        case Devourers_ID:
            p = Devourers();
            break;

        case Robbers_ID:
            p = Robbers();
            break;

        default:
            std::cout << "Ошибка\n";
    }
    return p;
};

void Warrior::SetPlaceX(int placex) {
    this->placex = placex;
}

int Warrior::GetPlaceX(void) {
    return placex;
}
```

```

void Warrior::SetPlaceY(int placey) {
    this->placey = placey;
}

int Warrior::GetPlaceY(void) {
    return placey;
}

void Warrior::SetValue(std::string value) {
    this->value = value;
}

std::string Warrior::GetValue(void) {
    return value;
}

/*void Health::SetHealth(int health) {
    this->health = health;
}

int Health::GetHealth() {
    return health;
}

void Damage::SetDamage(int damage) {
    this->damage = damage;
}

int Damage::GetDamage() {
    return damage;
}

void Armor::SetArmor(int armor) {
    this->armor = armor;
}

int Armor::GetArmor() {
    return armor;
}*/

void Warrior::SetHealth(int health) {
    this->health = health;
}

int Warrior::GetHealth() {
    return health;
}

void Warrior::SetDamage(int damage) {
    this->damage = damage;
}

int Warrior::GetDamage() {
    return damage;
}

void Warrior::SetArmor(int armor) {
    this->armor = armor;
}

int Warrior::GetArmor() {
    return armor;
}

```

## Приложение Е

### Содержимое файла Orcs.h

```
#pragma once
#include "Warrior.h"

class Orcs : public Warrior {
public:
    Orcs();
    ~Orcs();
    void SetBoost(int boost);

private:
    int boost;           //умножает здоровье и атаку на значение boost, а броню делит на
    (boost + 1)

};

//-----

class Devourers : public Orcs {           //пожиратели
public:
    Devourers();
    ~Devourers();
    int GetCounter();
    void SetCounter(int counter);

private:
    double percent;      //с вероятностью x при атаке урон умножается на 2
    int counter;

};

//-----

class Robbers : public Orcs {           //разбойники
public:
    Robbers();
    ~Robbers();
    int GetCounter();
    void SetCounter(int counter);

private:
    int around;          //если разбойник окружен (в упор к нему стоят минимум 2
    врага), то он наносит урон обоим
    int counter;

};
```

## Приложение Ж

### Содержимое файла Orcs.cpp

```
#include "Orcs.h"

Orcs::Orcs() {
    this->boost = 2;
    this->SetHealth(this->GetHealth() * boost);
    this->SetDamage(this->GetDamage() * boost);
    this->SetArmor(this->GetArmor() / (boost + 1));
    this->SetValue("0");
}

Orcs::~Orcs() {

}

void Orcs::SetBoost(int boost) {
    this->boost = boost;
    this->SetHealth(this->GetHealth() * boost);
    this->SetDamage(this->GetDamage() * boost);
    this->SetArmor(this->GetArmor() / (boost + 1));
}

//-----

Devourers::Devourers() : Orcs() {
    this->percent = 0.5;
    this->SetValue("OD");
    this->counter = 0;
}

Devourers::~Devourers() {

}

int Devourers::GetCounter() {
    return counter;
}

void Devourers::SetCounter(int counter) {
    this->counter = counter;
}

//-----

Robbers::Robbers() : Orcs() {
    this->around = 1;
    this->SetValue("OR");
    this->counter = 0;
}

Robbers::~Robbers() {

}

int Robbers::GetCounter() {
    return counter;
}

void Robbers::SetCounter(int counter) {
    this->counter = counter;
}
```

## Приложение 3

### Содержимое файла Men.h

```
#pragma once
#include "Warrior.h"

class Men : public Warrior {
public:
    Men();
    ~Men();
    void SetBoost(int boost);

private:
    int boost;                //умножает урон и броню на boost, здоровье делит на
boost

};

//-----

class Spearman : public Men {    //копьеносец
public:
    Spearman();
    ~Spearman();
    int GetCounter();
    void SetCounter(int counter);

private:
    int barbecue;            //если за врагом сразу же стоит еще один враг, то урон
наносится обоим
    int counter;
};

//-----

class Wizard : public Men {    // маг
public:
    Wizard();
    ~Wizard();
    int GetCounter();
    void SetCounter(int counter);

private:
    int range;                //может нанести урон, поделенный на 3, врагу на любом расстоянии
    int counter;
};
```

## Приложение И

### Содержимое файла Men.cpp

```
#include "Men.h"

Men::Men() {
    this->boost = 2;
    this->SetHealth(this->GetHealth() / boost);
    this->SetDamage(this->GetDamage() * boost);
    this->SetArmor(this->GetArmor() * boost);
    this->SetValue("M");
}

Men::~Men() {

}

void Men::SetBoost(int boost) {
    this->boost = boost;
    this->SetHealth(this->GetHealth() / boost);
    this->SetDamage(this->GetDamage() * boost);
    this->SetArmor(this->GetArmor() * boost);
}

//-----

Spearman::Spearman() : Men() {
    this->barbecue = 1;
    this->SetValue("MS");
    this->counter = 0;
}

Spearman::~Spearman() {

}

int Spearman::GetCounter() {
    return counter;
}

void Spearman::SetCounter(int counter) {
    this->counter = counter;
}

//-----

Wizard::Wizard() : Men() {
    this->range = 1;
    this->SetDamage(this->GetDamage() / 3);
    this->SetValue("MW");
    this->counter = 0;
}

Wizard::~Wizard() {

}

int Wizard::GetCounter() {
    return counter;
}

void Wizard::SetCounter(int counter) {
    this->counter = counter;
}
```

}



## Приложение К

### Содержимое файла Elfs.h

```
#pragma once
#include "Warrior.h"

class Elfs : public Warrior {
public:
    Elfs();
    ~Elfs();
    void SetBoost(int boost);

private:
    int boost;                //умножает урон на boost + 1, броню делит на boost,
    здоровье оставляет неизменным
};

//-----

class Archer : public Elfs {    //лучник
public:
    Archer();
    ~Archer();
    int GetCounter();
    void SetCounter(int counter);

private:
    int arr;                  //наносит урон врагу, находящегося максимум через 2 клетки
    от лучника
    int counter;
};

//-----

class Invisible : public Elfs {    //невидимка
public:
    Invisible();
    ~Invisible();
    int GetCounter();
    void SetCounter(int counter);

private:
    int invis;                //становится невидимым на 3 хода (через него нельзя шагать и
    на него нельзя нападать)
    int counter;
};
```

## Приложение Л

### Содержимое файла Elfs.cpp

```
#include "Elfs.h"

Elfs::Elfs() {
    this->boost = 2;
    this->SetDamage(this->GetDamage() * boost);
    this->SetArmor(this->GetArmor() / boost);
    this->SetValue("E");
}

Elfs::~Elfs() {

}

void Elfs::SetBoost(int boost) {
    this->boost = boost;
    this->SetDamage(this->GetDamage() * boost);
    this->SetArmor(this->GetArmor() / boost);
}

//-----

Archer::Archer() : Elfs() {
    this->arr = 2;
    this->SetValue("EA");
    this->counter = 0;
}

Archer::~Archer() {

}

int Archer::GetCounter() {
    return counter;
}

void Archer::SetCounter(int counter) {
    this->counter = counter;
}

//-----

Invisible::Invisible() : Elfs() {
    this->invis = 3;
    this->SetValue("EI");
    this->counter = 0;
}

Invisible::~Invisible() {

}

int Invisible::GetCounter() {
    return counter;
}

void Invisible::SetCounter(int counter) {
    this->counter = counter;
}
```

## Приложение М

### Содержимое файла Iterator.h

```
#pragma once

#include "Warrior.h"

class Iterator {
public:
    Iterator(Warrior** cell, int x, int y);
    void first();
    void last();
    void next();
    Warrior* getCell();

private:
    int x, y;
    int maxX, maxY;
    bool isEnd;
    Warrior** cell;
};
```

## Приложение Н

### Содержимое файла Iterator.cpp

```
#include "Iterator.h"

using namespace std;

Iterator::Iterator(Warrior** cell, int x, int y) {
    this->cell = cell;
    this->x = 0;
    this->y = 0;
    maxX = x - 1;
    maxY = y - 1;
    isEnd = false;
}

void Iterator::first() {
    x = 0;
    y = 0;

    cout << x << " " << y << endl;
}

void Iterator::last() {
    x = maxX;
    y = maxY;
    cout << x << " " << y << endl;
}

void Iterator::next() {
    if (x == maxX && y == maxY) isEnd = true;
    else if (x == maxX) {
        x = 0;
        y++;
        cout << x << " " << y << endl;
    }
    else {
        x++;
        cout << x << " " << y << endl;
    }
}

Warrior* Iterator::getCell() {
    if (isEnd) return NULL;
    return &cell[x][y];
}
```