Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №2

Вариант 2626

Выполнил:

Шмунк Андрей Александрович

Группа P3108

Проверил:

Преподаватель программирования

Харитонова Анастасия Евгеньевна

Содержание

[Задание 3](#_Toc148225752)

[UML-диаграмма классов 5](#_Toc148225753)

[Исходный код программы 5](#_Toc148225754)

[Атаки 5](#_Toc148225755)

[ShadowPunch 5](#_Toc148225756)

[Eruption 6](#_Toc148225757)

[HydroPump 6](#_Toc148225758)

[RazorWind 6](#_Toc148225759)

[Thunderbolt 7](#_Toc148225760)

[ThunderShock 7](#_Toc148225761)

[Camouflage 8](#_Toc148225762)

[LightScreen 8](#_Toc148225763)

[ThunderWave 8](#_Toc148225764)

[Покемоны 9](#_Toc148225765)

[Dratini 9](#_Toc148225766)

[Zigzagoon 9](#_Toc148225767)

[Linoone 9](#_Toc148225768)

[Magnemite 10](#_Toc148225769)

[Magneton 10](#_Toc148225770)

[Baltoy 10](#_Toc148225771)

[Claydol 11](#_Toc148225772)

[Cresselia 11](#_Toc148225773)

[Main-класс 12](#_Toc148225774)

[Результат работы программы 12](#_Toc148225775)

[Вывод 14](#_Toc148225776)

# Задание

На основе базового класса Pokemon написать свои классы для заданных видов покемонов. Каждый вид покемона должен иметь один или два типа и стандартные базовые характеристики:

* очки здоровья (HP)
* атака (attack)
* защита (defense)
* специальная атака (special attack)
* специальная защита (special defense)
* скорость (speed)

Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов PhysicalMove, SpecialMove и StatusMove реализовать свои классы для заданных видов атак.

Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

Используя класс симуляции боя Battle, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

Базовые классы и симулятор сражения находятся в [jar-архиве](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](https://se.ifmo.ru/~tony/doc/).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах <http://poke-universe.ru>, <http://pokemondb.net>,<http://veekun.com/dex/pokemon>

#### **Комментарии**

Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

Что надо сделать (краткое описание)

1. Ознакомиться с документацией, обращая особое внимание на классы Pokemon и Move. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
2. Скачать файл Pokemon.jar. Его необходимо будет использовать как для компиляции, так и для запуска программы. Распаковывать его не надо! Нужно научиться подключать внешние jar-файлы к своей программе.
3. Написать минимально работающую программу и посмотреть как она работает.

Battle b = new Battle();

Pokemon p1 = new Pokemon("Чужой", 1);

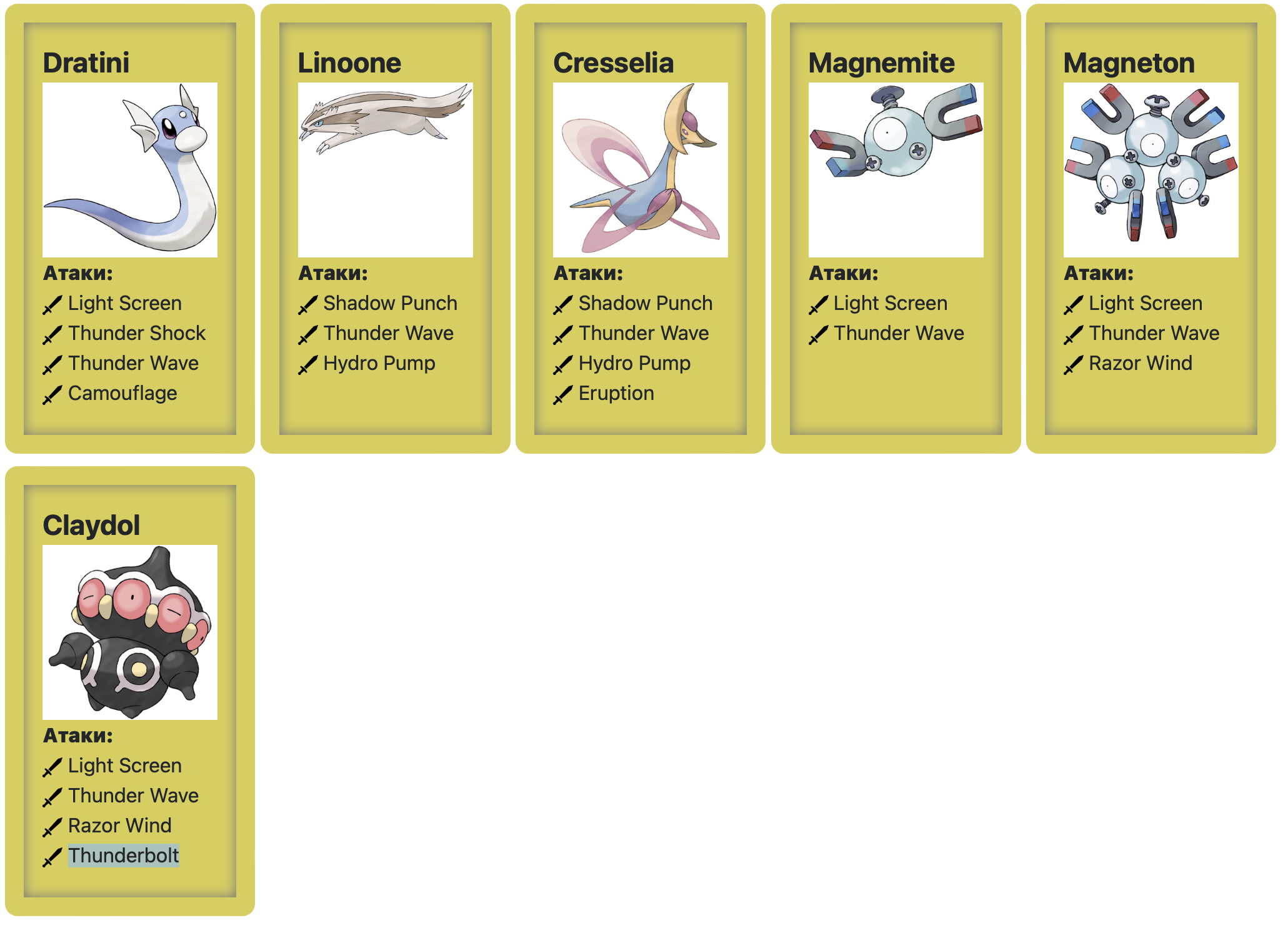
Pokemon p2 = new Pokemon("Хищник", 1);

b.addAlly(p1);

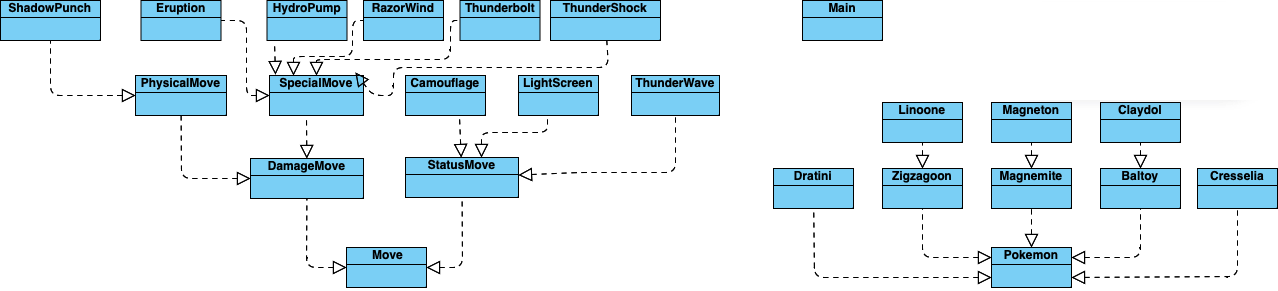
b.addFoe(p2);

b.go();

1. Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса Pokemon. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.
2. Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса PhysicalMove или SpecialMove. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаку покемону и проверить ее действие в сражении. Не забудьте переопределить метод describe, чтобы выводилось нужное сообщение.
3. Если действие атаки отличается от стандартного, например, покемон не промахивается, либо атакующий покемон также получает повреждение, то в классе атаки нужно дополнительно переопределить соответствующие методы (см. документацию). При реализации атак, которые меняют статус покемона (наследники StatusMove), скорее всего придется разобраться с классом Effect. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.
4. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.



# UML-диаграмма классов



# Исходный код программы

## Атаки

### ShadowPunch

package attacks.physical;

import ru.ifmo.se.pokemon.\*;

public class ShadowPunch extends PhysicalMove{

public ShadowPunch(){

super(Type.GHOST,60,100);

}

protected String describe(){

return "использует атаку Shadow Punch";

}

}

### Eruption

package attacks.special;

import ru.ifmo.se.pokemon.\*;

public class Eruption extends SpecialMove{

public Eruption(){

super(Type.FIRE,150,100);

}

protected String describe(){

return "использует атаку Eruption";

}

protected double calcBaseDamage(Pokemon att, Pokemon def){

return 150\*def.getHP()/def.getStat(Stat.HP);

}

}

### HydroPump

package attacks.special;

import ru.ifmo.se.pokemon.\*;

public class HydroPump extends SpecialMove{

public HydroPump(){

super(Type.WATER,110,80);

}

protected String describe(){

return "использует атаку Hydro Pump";

}

}

### RazorWind

package attacks.special;

import ru.ifmo.se.pokemon.\*;

public class RazorWind extends SpecialMove{

public RazorWind(){

super(Type.NORMAL,80,100);

}

protected String describe(){

return "использует атаку Razor Wind";

}

protected double calcCriticalHit(Pokemon pokemon, Pokemon pokemon1) {

if (1./8. > Math.random()) {

System.out.println("Критический удар!");

return 2.0;

}

else {

return 1.0;

}

}

}

### Thunderbolt

package attacks.special;

import ru.ifmo.se.pokemon.\*;

public class Thunderbolt extends SpecialMove{

public Thunderbolt(){

super(Type.ELECTRIC,90,100);

}

protected String describe(){

return "использует атаку Thunderbolt";

}

protected void applyOppEffects(Pokemon p) {

p.setCondition(new Effect().chance(0.1).condition(Status.PARALYZE));

}

}

### ThunderShock

package attacks.special;

import ru.ifmo.se.pokemon.\*;

public class ThunderShock extends SpecialMove{

public ThunderShock(){

super (Type.ELECTRIC,40,100);

}

protected String describe(){

return "использует атаку Thunder Shock";

}

protected void applyOppEffects(Pokemon p){

p.setCondition(new Effect().chance(0.1).condition(Status.PARALYZE));

}

}

### Camouflage

package attacks.status;

import ru.ifmo.se.pokemon.\*;

public class Camouflage extends StatusMove{

public Camouflage(){

super(Type.NORMAL,0,0);

}

protected String describe(){

return "использует атаку Camouflage";

}

}

### LightScreen

package attacks.status;

import ru.ifmo.se.pokemon.\*;

public class LightScreen extends StatusMove{

public LightScreen() {

super(Type.PSYCHIC,0,0);

}

protected String describe(){

return "использует атаку Light Screen";

}

}

### ThunderWave

package attacks.status;

import ru.ifmo.se.pokemon.\*;

public class ThunderWave extends StatusMove{

public ThunderWave(){

super(Type.ELECTRIC,0,90);

}

protected String describe(){

return "Использует Thunder Wave";

}

protected void applyOppEffects(Pokemon p){

p.setCondition(new Effect().condition(Status.PARALYZE));

}

}

## Покемоны

### Dratini

package pokemon;

import ru.ifmo.se.pokemon.\*;

import attacks.status.\*;

import attacks.special.\*;

public class Dratini extends Pokemon {

public Dratini(){

this("Безымянный",1);

}

public Dratini(String name, int level){

super(name,level);

this.setType(Type.DRAGON);

this.setStats(41,64,45,50,50,50);

this.setMove(new LightScreen(), new ThunderShock(),new ThunderWave());

}

}

### Zigzagoon

package pokemon;

import ru.ifmo.se.pokemon.\*;

public class Zigzagoon extends Pokemon{

public Zigzagoon(){

this("Безымянный",1);

}

public Zigzagoon (String name, int level){

super(name,level);

this.setType(Type.NORMAL);

this.setStats(38,30,41,30,41,60);

}

}

### Linoone

package pokemon;

import ru.ifmo.se.pokemon.\*;

import attacks.special.\*;

import attacks.physical.\*;

import attacks.status.\*;

public class Linoone extends Zigzagoon{

public Linoone(){

this("Безымянный",1);

}

public Linoone(String name,int level){

super(name,level);

this.setStats(78,70,61,50,61,100);

this.setType(Type.NORMAL);

this.setMove(new ShadowPunch(), new ThunderWave(), new HydroPump());

}

}

### Magnemite

package pokemon;

import ru.ifmo.se.pokemon.\*;

import attacks.status.\*;

public class Magnemite extends Pokemon{

public Magnemite(){

this("Безымянный",1);

}

public Magnemite(String name,int level){

super(name,level);

this.setType(Type.ELECTRIC, Type.STEEL);

this.setStats(25,35,70,95,55,45);

this.setMove(new LightScreen(), new ThunderWave());

}

}

### Magneton

package pokemon;

import ru.ifmo.se.pokemon.\*;

import attacks.status.\*;

import attacks.special.\*;

public class Magneton extends Magnemite{

public Magneton(){

this("Безымянный",1);

}

public Magneton(String name, int level){

super(name,level);

this.setType(Type.ELECTRIC, Type.STEEL);

this.setStats(50,60,95,120,70,70);

this.addMove(new RazorWind());

}

}

### Baltoy

package pokemon;

import ru.ifmo.se.pokemon.\*;

public class Baltoy extends Pokemon{

public Baltoy(){

this("Безымянный",1);

}

public Baltoy(String name,int level){

super(name,level);

this.setType(Type.GROUND, Type.PSYCHIC);

this.setStats(40,40,55,40,70,55);

}

}

### Claydol

package pokemon;

import ru.ifmo.se.pokemon.\*;

import attacks.status.\*;

import attacks.special.\*;

public class Claydol extends Baltoy{

public Claydol(){

this("Безымянный",1);

}

public Claydol(String name,int level){

super(name,level);

this.setType(Type.GROUND, Type.PSYCHIC);

this.setStats(60,70,105,70,120,75);

this.setMove(new LightScreen(), new ThunderWave(), new RazorWind(), new Thunderbolt());

}

}

### Cresselia

package pokemon;

import ru.ifmo.se.pokemon.\*;

import attacks.special.\*;

import attacks.status.\*;

import attacks.physical.\*;

public class Cresselia extends Pokemon{

public Cresselia(){

this("Безымянный",1);

}

public Cresselia(String name,int level){

super(name,level);

this.setType(Type.PSYCHIC);

this.setStats(120,70,110,75,120,85);

this.setMove(new ShadowPunch(), new ThunderWave(), new HydroPump(), new Eruption());

}

}

## Main-класс

import ru.ifmo.se.pokemon.\*;

import pokemon.\*;

public class Main {

public static void main(String[] args) {

Battle b = new Battle();

Dratini p1 = new Dratini("Змейка",2);

Linoone p2 = new Linoone("Летяга",3);

Cresselia p3 = new Cresselia("Фея",1);

Magnemite p4 = new Magnemite("Притягиваетль",2);

Magneton p5 = new Magneton("Большая притягивалка",2);

Claydol p6 = new Claydol("Инжир",3);

b.addAlly(p1);

b.addAlly(p3);

b.addAlly(p5);

b.addFoe(p2);

b.addFoe(p4);

b.addFoe(p6);

b.go();

}

}

# Результат работы программы

Dratini Змейка из команды синих вступает в бой!

Linoone Летяга из команды полосатых вступает в бой!

Linoone Летяга использует атаку Shadow Punch.

Dratini Змейка теряет 4 здоровья.

Dratini Змейка использует атаку Thunder Shock.

Linoone Летяга теряет 4 здоровья.

Получен урон от электричества

Linoone Летяга использует атаку Hydro Pump.

Dratini Змейка теряет 3 здоровья.

Dratini Змейка промахивается

Linoone Летяга Использует Thunder Wave.

Dratini Змейка парализован

Dratini Змейка промахивается

Linoone Летяга использует атаку Hydro Pump.

Dratini Змейка теряет 3 здоровья.

Dratini Змейка использует атаку Thunder Shock.

Linoone Летяга теряет 5 здоровья.

Получен урон от электричества

Linoone Летяга использует атаку Shadow Punch.

Dratini Змейка теряет 4 здоровья.

Dratini Змейка теряет сознание.

Cresselia Фея из команды синих вступает в бой!

Linoone Летяга использует атаку Shadow Punch.

Cresselia Фея теряет 8 здоровья.

Cresselia Фея использует атаку Hydro Pump.

Linoone Летяга теряет 6 здоровья.

Linoone Летяга использует атаку Shadow Punch.

Cresselia Фея теряет 14 здоровья.

Cresselia Фея теряет сознание.

Magneton Большая притягивалка из команды синих вступает в бой!

Linoone Летяга Использует Thunder Wave.

Magneton Большая притягивалка парализован

Magneton Большая притягивалка использует атаку Razor Wind.

Linoone Летяга теряет 5 здоровья.

Linoone Летяга теряет сознание.

Magnemite Притягиваетль из команды полосатых вступает в бой!

Magneton Большая притягивалка Использует Thunder Wave.

Magnemite Притягиваетль парализован

Magnemite Притягиваетль промахивается

Magneton Большая притягивалка промахивается

Magnemite Притягиваетль промахивается

Magneton Большая притягивалка Использует Thunder Wave.

Magnemite Притягиваетль Использует Thunder Wave.

Magneton Большая притягивалка Использует Thunder Wave.

Magnemite Притягиваетль Использует Thunder Wave.

Magneton Большая притягивалка использует атаку Razor Wind.

Критический удар!

Magnemite Притягиваетль теряет 7 здоровья.

Magnemite Притягиваетль Использует Thunder Wave.

Magneton Большая притягивалка Использует Thunder Wave.

Magnemite Притягиваетль Использует Thunder Wave.

Magneton Большая притягивалка использует атаку Razor Wind.

Magnemite Притягиваетль теряет 4 здоровья.

Magnemite Притягиваетль промахивается

Magneton Большая притягивалка использует атаку Razor Wind.

Magnemite Притягиваетль теряет 3 здоровья.

Magnemite Притягиваетль теряет сознание.

Claydol Инжир из команды полосатых вступает в бой!

Claydol Инжир использует атаку Razor Wind.

Критический удар!

Magneton Большая притягивалка теряет 7 здоровья.

Magneton Большая притягивалка промахивается

Claydol Инжир использует атаку Razor Wind.

Magneton Большая притягивалка теряет 3 здоровья.

Magneton Большая притягивалка использует атаку Razor Wind.

Claydol Инжир теряет 6 здоровья.

Claydol Инжир промахивается

Magneton Большая притягивалка промахивается

Claydol Инжир использует атаку Thunderbolt.

Magneton Большая притягивалка теряет 4 здоровья.

получен урон от электричества

Magneton Большая притягивалка промахивается

Claydol Инжир использует атаку Thunderbolt.

Magneton Большая притягивалка теряет 4 здоровья.

получен урон от электричества

Magneton Большая притягивалка теряет сознание.

В команде синих не осталось покемонов.

Команда полосатых побеждает в этом бою!

# Вывод

В ходе данной работы я научился подключать внешние jar файлы и использовать их. Научился работать с документацией, изучил объектно-ориентированный подход на языке Java. Научился работать с классами, конструкторами, полями и модификаторами доступа.