Университет ИТМО Мегафакультет компьютерных

технологий и управления

Факультет программной инженерии и

компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА № 2

по дисциплине

‘ПРОГРАММИРОВАНИЕ’

Вариант 90556

Выполнил: Киеу Чыонг Занг

Группа: P3107

Преподаватель: Гаврилов Антон Валерьевич

Г.Санкт-Петербург

2022

Table of Contents

[Покемоны: 3](#_Toc117085116)

[Диаграмма классов: 4](#_Toc117085117)

[Исходный код: 5](#_Toc117085118)

[Battleground.java 5](#_Toc117085119)

[moves/ 5](#_Toc117085120)

[DazzlingGleam.java 5](#_Toc117085121)

[DefenseCurl.java 5](#_Toc117085122)

[FocusBlast.java 6](#_Toc117085123)

[PoisonJab.java 6](#_Toc117085124)

[Rest.java 6](#_Toc117085125)

[ShadowClaw.java 7](#_Toc117085126)

[SwordsDance.java 7](#_Toc117085127)

[Thunderbolt.java 7](#_Toc117085128)

[ThunderWave.java 8](#_Toc117085129)

[WillOWisp.java 8](#_Toc117085130)

[WorkUp.java 8](#_Toc117085131)

[pokemons/ 9](#_Toc117085132)

[Darkrai.java 9](#_Toc117085133)

[Minccino.java 9](#_Toc117085134)

[Cinccino.java 9](#_Toc117085135)

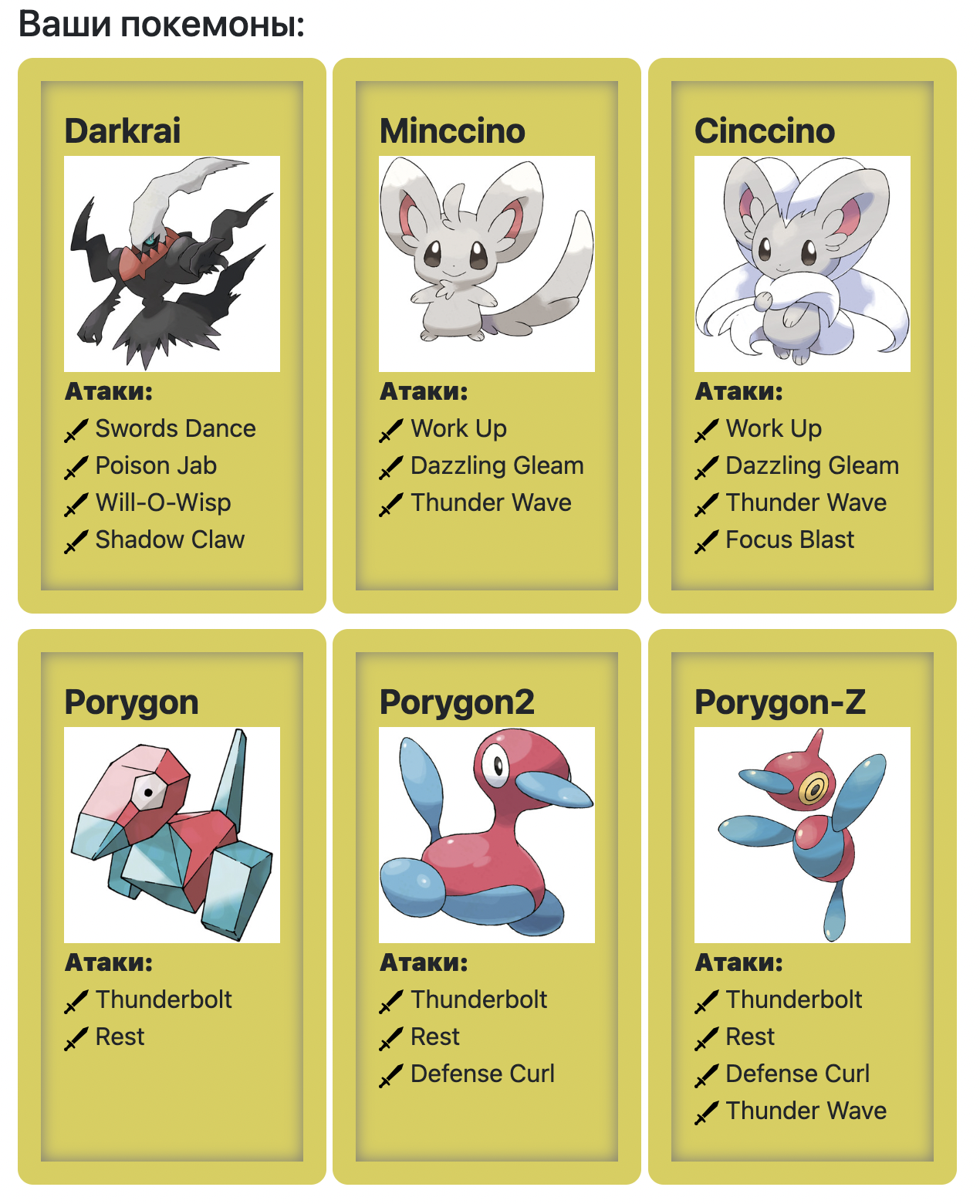
[Porygon.java 10](#_Toc117085136)

[Porygon2.java 10](#_Toc117085137)

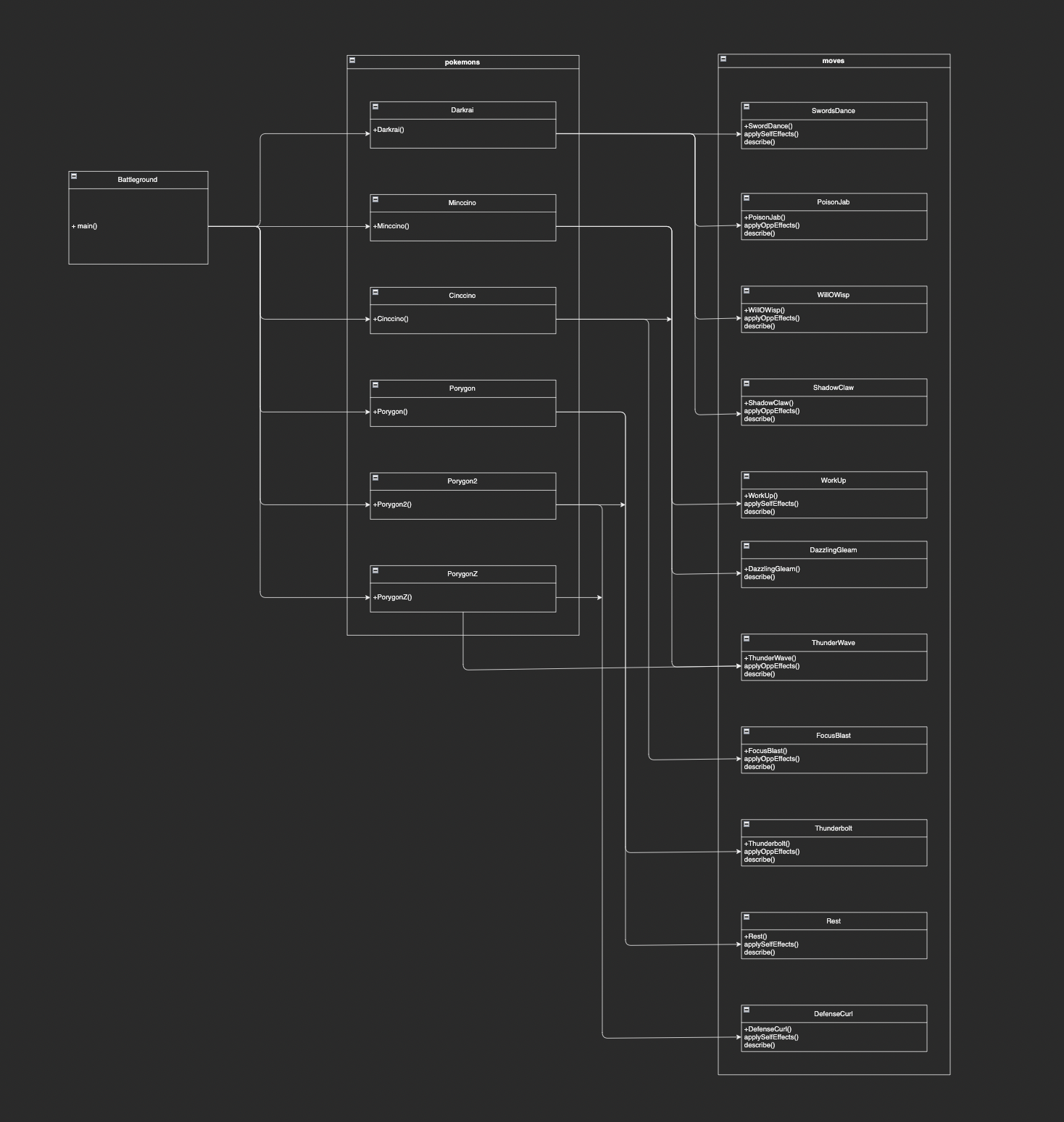
[PorygonZ.java 10](#_Toc117085138)

[Вывод: 14](#_Toc117085139)

# Покемоны:



# Диаграмма классов:



# Исходный код:

## Battleground.java

import pokemons.\*;  
import ru.ifmo.se.pokemon.\*;  
  
class Battleground {  
 public static void main(String args[]) {  
 Battle b = new Battle();  
  
 Minccino p1= new Minccino ("ally", 1);  
 Cinccino p2 = new Cinccino ("ally", 1);  
 Darkrai p3 = new Darkrai ("ally", 1);  
  
 Porygon p4= new Porygon ("foe", 1);  
 Porygon2 p5= new Porygon2 ("foe", 1);  
 PorygonZ p6 = new PorygonZ ("foe", 1);  
  
 b.addAlly(p1);  
 b.addAlly(p2);  
 b.addAlly(p3);  
 b.addFoe(p4);  
 b.addFoe(p5);  
 b.addFoe(p6);  
 b.go();  
 }  
}

## moves/

### DazzlingGleam.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DazzlingGleam extends SpecialMove {  
 public DazzlingGleam(){ super(Type.*FAIRY*, 80, 100);}  
  
 @Override  
 protected String describe(){  
 return "uses Dazzling Gleam.";  
 }  
}

### DefenseCurl.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DefenseCurl extends StatusMove {  
 public DefenseCurl() {super(Type.*NORMAL*, 0, 0);}  
  
 @Override  
 protected void applySelfEffects(Pokemon p){  
 p.setMod(Stat.*DEFENSE*, 1);  
 }  
  
 @Override  
 protected String describe(){ return "uses Defense Curl.";}  
}

### FocusBlast.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class FocusBlast extends SpecialMove {  
 public FocusBlast(){ super(Type.*FIGHTING*, 120, 70);}  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if(Math.*random*() < 0.1){  
 p.setMod(Stat.*SPECIAL\_DEFENSE*, -1);  
 }  
 }  
 @Override  
 protected String describe(){  
 return "uses Focus Blast.";  
 }  
}

### PoisonJab.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class PoisonJab extends PhysicalMove {  
 public PoisonJab() { super(Type.*POISON*, 80, 100);}  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if (Math.*random*() < 0.3){  
 Effect.*poison*(p);  
 }  
 }  
 @Override  
 protected String describe() {  
 return "uses Poison Jab";  
 }  
}

### Rest.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Rest extends StatusMove {  
 public Rest() {  
 super(Type.*PSYCHIC*, 0, 0);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 Effect e1 = new Effect().chance(1).turns(0).stat(Stat.*HP*, -343749);  
 Effect e2 = new Effect().chance(1).turns(2).condition(Status.*SLEEP*);

p.addEffect(e1);  
 p.setCondition(e2);  
 }  
 @Override  
 protected String describe() {  
 return "uses Rest";  
 }  
}

### ShadowClaw.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class ShadowClaw extends PhysicalMove {  
 public ShadowClaw(){ super(Type.*GHOST*, 70, 100);}  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if(Math.*random*() < 0.1){  
 p.setMod(Stat.*SPECIAL\_DEFENSE*, -1);  
 }  
 }  
 @Override  
 protected String describe(){  
 return "uses Shadow Claw.";  
 }  
  
}

### SwordsDance.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class SwordsDance extends StatusMove {  
 public SwordsDance(){super(Type.*NORMAL*, 0, 0);}  
  
 @Override  
 protected void applySelfEffects(Pokemon p){  
 p.setMod(Stat.*ATTACK*, 2);  
 }  
  
 @Override  
 protected String describe(){  
 return "uses Swords Dance.";  
 }  
  
}

### Thunderbolt.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Thunderbolt extends SpecialMove {  
 public Thunderbolt(){ super(Type.*ELECTRIC*, 90, 100);}  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if(Math.*random*() < 0.1){  
 Effect.*paralyze*(p);  
 }  
 }  
 @Override  
 protected String describe(){  
 return "uses Thunderbolt.";  
 }  
}

### ThunderWave.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class ThunderWave extends StatusMove {  
 public ThunderWave() {super(Type.*ELECTRIC*, 0, 90);}  
  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 Effect.*paralyze*(p);  
 }  
  
 @Override  
 protected String describe() {  
 return "uses Thunder Wave";  
 }  
}

### WillOWisp.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class WillOWisp extends StatusMove {  
 public WillOWisp(){ super(Type.*FIRE*, 0,85);}  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 Effect.*burn*(p);  
 }  
  
 @Override  
 protected String describe(){  
 return "uses Will-O-Wisp";  
 }  
}

### WorkUp.java

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class WorkUp extends StatusMove {  
 public WorkUp(){ super(Type.*NORMAL*, 0, 0);}  
  
 @Override  
 protected void applySelfEffects(Pokemon p){  
 p.setMod(Stat.*ATTACK*, 1);  
 p.setMod(Stat.*SPECIAL\_ATTACK*, 1);  
 }  
 @Override  
 protected String describe(){  
 return "uses Work Up.";  
 }  
  
}

## pokemons/

### Darkrai.java

package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Darkrai extends Pokemon {  
 public Darkrai(String name, int level){  
 super(name, level);  
 setType(Type.*DARK*);  
 setStats(70, 90, 90, 135, 90, 125);  
 setMove(new SwordsDance(), new PoisonJab(), new WillOWisp(), new ShadowClaw());  
  
 }  
}

### Minccino.java

package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Minccino extends Pokemon {  
 public Minccino(String name, int level){  
 super(name, level);  
 setType(Type.*NORMAL*);  
 setStats(55, 50, 40, 40, 40, 75);  
 setMove(new WorkUp(), new DazzlingGleam(), new ThunderWave());  
 }  
}

### Cinccino.java

package pokemons;  
  
import moves.FocusBlast;  
import pokemons.Minccino;  
  
public class Cinccino extends Minccino{  
 public Cinccino(String name, int level){  
 super(name, level);  
 setStats(75, 95, 60, 65, 60, 115);  
 addMove(new FocusBlast());  
 }  
}

### Porygon.java

package pokemons;  
  
import moves.Rest;  
import moves.Thunderbolt;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Porygon extends Pokemon {  
 public Porygon(String name, int level){  
 super(name, level);  
 setType(Type.*NORMAL*);  
 setStats(65, 60, 70, 85, 75, 40);  
 setMove(new Thunderbolt(), new Rest());  
 }  
}

### Porygon2.java

package pokemons;  
  
import moves.DefenseCurl;  
import pokemons.Porygon;  
public class Porygon2 extends Porygon{  
 public Porygon2(String name, int level){  
 super(name, level);  
 setStats(85, 80, 90, 105, 95, 60);  
 addMove(new DefenseCurl());  
 }  
}

### PorygonZ.java

package pokemons;  
  
import moves.ThunderWave;  
import pokemons.Porygon2;  
  
public class PorygonZ extends Porygon2 {  
 public PorygonZ(String name, int level){  
 super(name, level);  
 setStats(85, 80, 70, 135, 75, 90);  
 addMove(new ThunderWave());  
 }  
}

# Результат работы программы:

Minccino ally from the team Purple enters the battle!

Porygon foe from the team Striped enters the battle!

Minccino ally uses Thunder Wave.

Porygon foe is paralyzed

Porygon foe misses

Porygon foe uses Thunderbolt..

Critical hit!

Minccino ally loses 11 hit points.

Minccino ally uses Thunder Wave.

Porygon foe misses

Minccino ally misses

Porygon foe uses Thunderbolt..

Critical hit!

Minccino ally loses 14 hit points.

Minccino ally faints.

Cinccino ally from the team Purple enters the battle!

Porygon foe uses Thunderbolt..

Critical hit!

Cinccino ally loses 13 hit points.

Cinccino ally faints.

Darkrai ally from the team Purple enters the battle!

Porygon foe misses

Darkrai ally uses Shadow Claw..

Porygon foe loses 1 hit points.

Porygon foe isn't affected by GHOST

Porygon foe misses

Darkrai ally misses

Porygon foe misses

Darkrai ally uses Poison Jab.

Porygon foe loses 6 hit points.

Porygon foe misses

Darkrai ally uses Will-O-Wisp.

Porygon foe is burned

Darkrai ally misses

Porygon foe uses Thunderbolt..

Darkrai ally loses 7 hit points.

Darkrai ally uses Shadow Claw..

Porygon foe loses 1 hit points.

Porygon foe isn't affected by GHOST

Porygon foe uses Thunderbolt..

Darkrai ally loses 5 hit points.

Darkrai ally uses Shadow Claw..

Porygon foe loses 1 hit points.

Porygon foe isn't affected by GHOST

Porygon foe uses Thunderbolt..

Darkrai ally loses 4 hit points.

Darkrai ally faints.

Team Purple loses its last Pokemon.

The team Striped wins the battle!

Process finished with exit code 0

# Вывод:

Во время выполнения данной лабораторной работы я познакомился с ООП, методами, классами, модификаторами доступами и сторонними библиотеками.

<https://github.com/ktgiang209/Prog-Labs/tree/main/Lab2>