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

Санкт-Петербург

2022

Table of Contents

[Диаграмма классов: 3](#_Toc115870966)

[Покемоны: 4](#_Toc115870967)

[Исходный код: 4](#_Toc115870968)

[Battleground.java 4](#_Toc115870969)

[moves/ 5](#_Toc115870970)

[Bubble.java 5](#_Toc115870971)

[Bulldoze.java 5](#_Toc115870972)

[DoubleTeam.java 6](#_Toc115870973)

[Facade.java 7](#_Toc115870974)

[FlameCharge.java 7](#_Toc115870975)

[Hypnosis.java 8](#_Toc115870976)

[IceBeam.java 8](#_Toc115870977)

[MetalSound.java 9](#_Toc115870978)

[PoisonJab.java 9](#_Toc115870979)

[Rest.java 10](#_Toc115870980)

[StoneEdge.java 11](#_Toc115870981)

[ThunderWave.java 12](#_Toc115870982)

[pokemons/ 12](#_Toc115870983)

[Mew.java 12](#_Toc115870984)

[Minior.java 13](#_Toc115870985)

[Poliwag.java 13](#_Toc115870986)

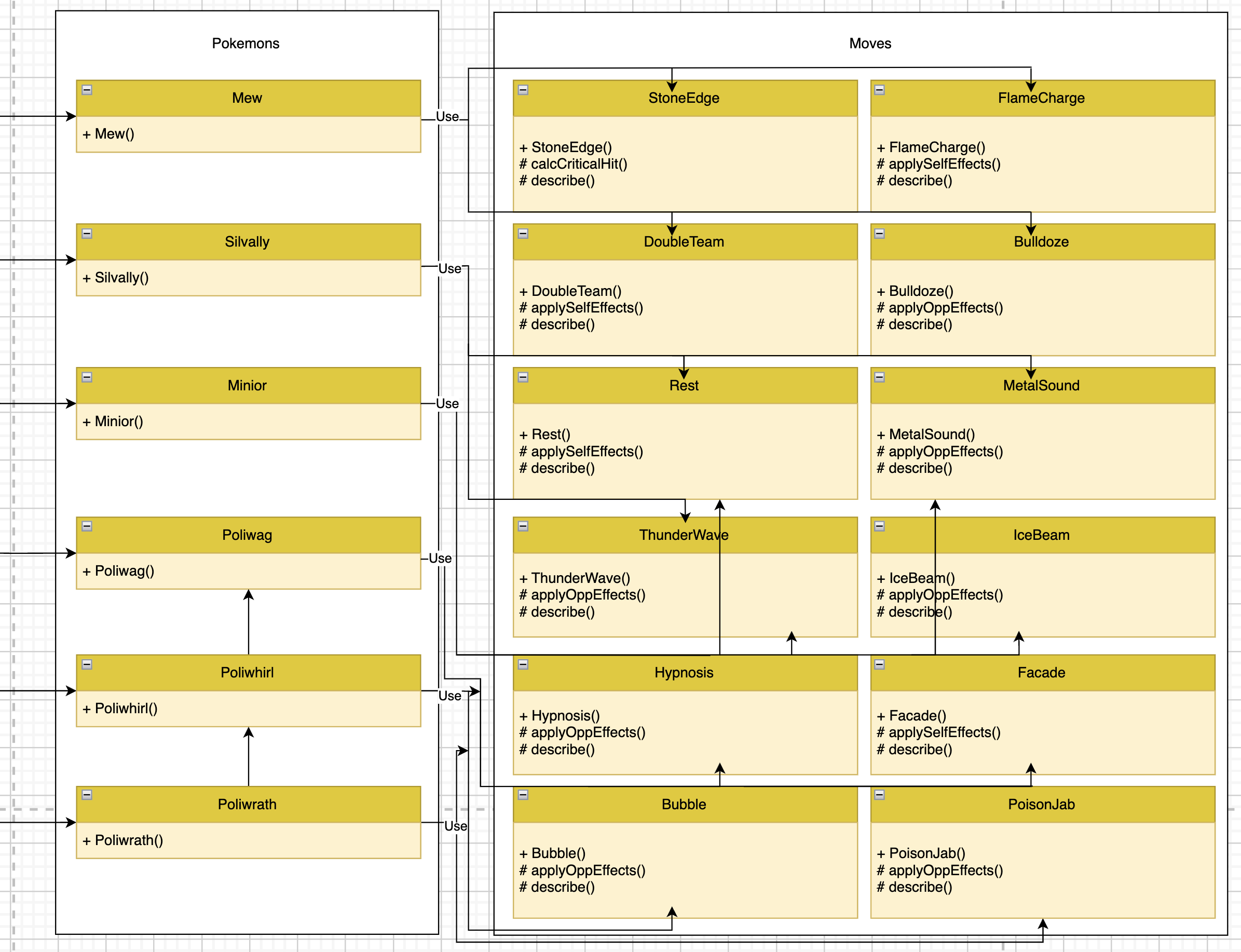
[Poliwhirl.java 14](#_Toc115870987)

[Poliwrath.java 14](#_Toc115870988)

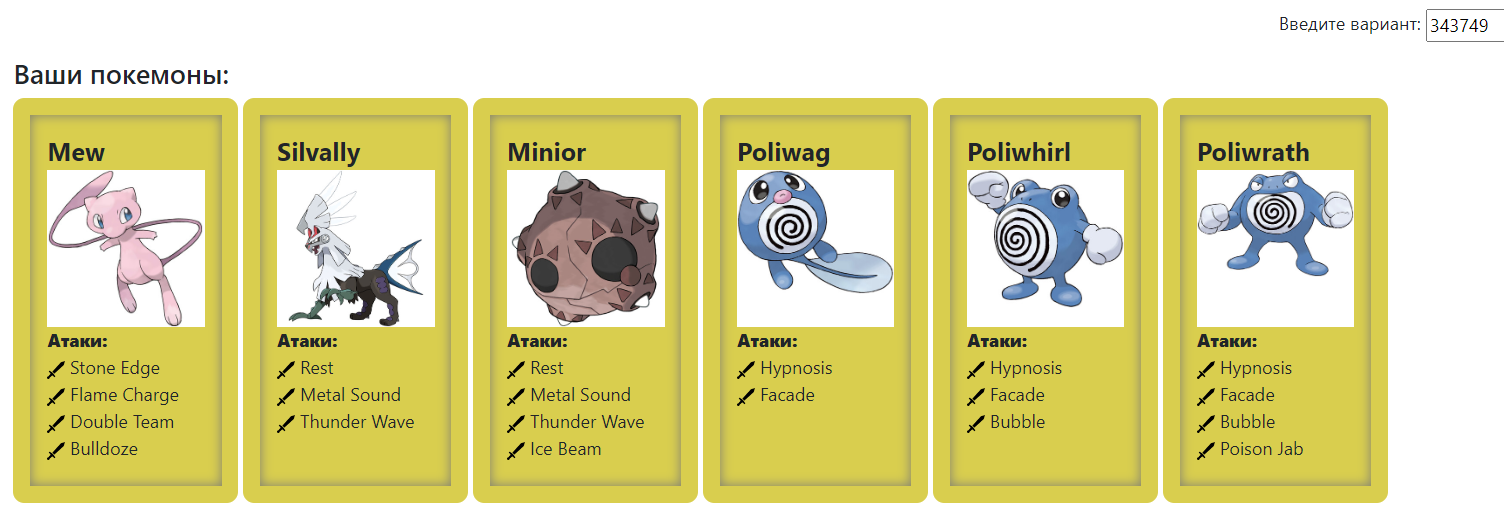
[Silvally.java 15](#_Toc115870989)

[Результат работы: 15](#_Toc115870990)

# Диаграмма классов:



# Покемоны:



# Исходный код:

## Battleground.java

import pokemons.\*;

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

class Battleground {

public static void main(String args[]) {

Battle b = new Battle();

Mew p1 = new Mew ("ally", 1);

Minior p2 = new Minior ("ally", 1);

Poliwag p3= new Poliwag ("ally", 1);

Poliwhirl p4= new Poliwhirl ("foe", 1);

Poliwrath p5= new Poliwrath ("foe", 1);

Silvally p6 = new Silvally ("foe", 1);

b.addAlly(p1);

b.addAlly(p2);

b.addAlly(p3);

b.addFoe(p4);

b.addFoe(p5);

b.addFoe(p6);

b.go();

}

}

## moves/

### **Bubble.java**

package moves;

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

public class Bubble extends SpecialMove {

public Bubble() { super(Type.WATER,40,100);}

@Override

protected void applyOppEffects(Pokemon p){

if(Math.random() < 0.1){

p.setMod(Stat.SPEED, 1);

}

}

@Override

protected String describe() {

return "using Bubble.\nThe target's Speed is lowering by one stage.";

}

}

### **Bulldoze.java**

package moves;

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

public class Bulldoze extends PhysicalMove {

public Bulldoze() { super(Type.GROUND, 60, 100);}

@Override

protected void applyOppEffects(Pokemon p){

p.setMod(Stat.SPEED,-1);

}

@Override

protected String describe() {

return "using Bulldoze";

}

}

### **DoubleTeam.java**

package moves;

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

public class DoubleTeam extends StatusMove {

public DoubleTeam() {

super(Type.NORMAL, 0, 0);

}

@Override

protected void applySelfEffects(Pokemon p) {

p.setMod(Stat.EVASION, 1);

}

@Override

protected String describe() {

return "using Double Team";

}

}

### **Facade.java**

package moves;

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

public class Facade extends PhysicalMove {

public Facade() {

super(Type.NORMAL, 70, 100);

}

@Override

protected void applySelfEffects(Pokemon p) {

Status p\_stat = p.getCondition();

if (p\_stat.equals(Status.BURN) || p\_stat.equals(Status.POISON)

|| p\_stat.equals(Status.PARALYZE)) {

power = 2\*power;

}

}

@Override

protected String describe() {

return "using Facade";

}

}

### **FlameCharge.java**

package moves;

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

public class FlameCharge extends PhysicalMove {

public FlameCharge() { super(Type.FIRE, 50, 100);}

@Override

protected void applySelfEffects(Pokemon p){

p.setMod(Stat.SPEED, 1);

}

@Override

protected String describe() {

return "using Flame Charge";

}

}

### **Hypnosis.java**

package moves;

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

public class Hypnosis extends StatusMove {

public Hypnosis() { super(Type.PSYCHIC, 0, 60);}

@Override

protected void applyOppEffects(Pokemon p){

Effect.sleep(p);

}

@Override

protected String describe() {

return "using Hypnosis";

}

}

### **IceBeam.java**

package moves;

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

public class IceBeam extends SpecialMove {

public IceBeam() {super(Type.ICE, 90, 100);}

@Override

protected void applyOppEffects(Pokemon p) {

if (Math.random() <0.1) {

Effect.freeze(p);

}

}

@Override

protected String describe() {

return "using IceBeam";

}

}

### **MetalSound.java**

package moves;

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

public class MetalSound extends StatusMove {

public MetalSound(){ super( Type.STEEL, 0, 85);}

@Override

protected void applyOppEffects(Pokemon p){

p.setMod(Stat.SPECIAL\_DEFENSE, -2);

}

@Override

protected String describe() {

return "using Metal Sound";

}

}

### **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);

}

}

protected String describe() {

return "using 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().turns(0).stat(Stat.HP, -343749);

Effect e2 = new Effect().turns(2).condition(Status.SLEEP);

p.addEffect(e1);

p.addEffect(e2);

}

@Override

protected String describe() {

return "using Rest";

}

}

### **StoneEdge.java**

package moves;

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

public class StoneEdge extends PhysicalMove {

public StoneEdge() {

super(Type.ROCK, 100, 80);

}

@Override

protected double calcCriticalHit(Pokemon att, Pokemon def){

if (Math.random() < (att.getStat(Stat.SPEED)\*3/512)) {

return 2.0;

}

else {

return 1.0;

}

}

@Override

protected String describe() {

return "using Stone Edge";

}

}

### **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 "using Thunder Wave";

}

}

## pokemons/

### **Mew.java**

package pokemons;

import moves.\*;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Mew extends Pokemon {

public Mew(String name, int level){

super(name, level);

setType(Type.PSYCHIC);

setStats(100, 100, 100, 100, 100, 100);

setMove(new StoneEdge(), new FlameCharge(), new DoubleTeam(), new Bulldoze());

}

}

### **Minior.java**

package pokemons;

import moves.\*;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Minior extends Pokemon {

public Minior(String name, int level) {

super(name, level);

setType(Type.ROCK, Type.FLYING);

setStats(60, 60, 100, 60, 100, 60);

setMove(new Rest(), new MetalSound(), new ThunderWave());

}

}

### **Poliwag.java**

package pokemons;

import moves.\*;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Poliwag extends Pokemon {

public Poliwag(String name, int level){

super(name, level);

setType(Type.WATER);

setStats(40, 50, 40, 40, 40, 90);

setMove(new Hypnosis(), new Facade());

}

}

### **Poliwhirl.java**

package pokemons;

import moves.Bubble;

public class Poliwhirl extends Poliwag {

public Poliwhirl(String name, int level){

super(name, level);

setStats(65, 65, 65, 50, 50, 90);

addMove(new Bubble());

}

}

### **Poliwrath.java**

package pokemons;

import moves.PoisonJab;

import ru.ifmo.se.pokemon.Type;

public class Poliwrath extends Poliwhirl{

public Poliwrath(String name, int level){

super(name, level);

addType(Type.FIGHTING);

setStats(90, 95, 95, 70,90, 70);

addMove(new PoisonJab());

}

}

### **Silvally.java**

package pokemons;

import moves.\*;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Silvally extends Pokemon {

public Silvally(String name, int level){

super(name, level);

setType(Type.NORMAL);

setStats(95, 95, 95, 95, 95, 95);

setMove(new Rest(), new MetalSound(), new ThunderWave());

}

}

# Результат работы:

Mew ally from the team Yellow enters the battle!

Poliwhirl foe from the team Blue enters the battle!

Mew ally misses

Poliwhirl foe using Bubble.

The target's Speed is lowering by one stage..

Mew ally loses 7 hit points.

Mew ally using Flame Charge.

Poliwhirl foe loses 3 hit points.

Mew ally increases speed.

Poliwhirl foe using Facade.

Mew ally loses 4 hit points.

Mew ally using Stone Edge.

Poliwhirl foe loses 4 hit points.

Poliwhirl foe using Facade.

Mew ally loses 4 hit points.

Mew ally faints.

Minior ally from the team Yellow enters the battle!

Poliwhirl foe using Facade.

Minior ally loses 2 hit points.

Minior ally misses

Poliwhirl foe using Facade.

Minior ally loses 2 hit points.

Minior ally misses

Poliwhirl foe using Facade.

Minior ally loses 2 hit points.

Minior ally using Metal Sound.

Poliwhirl foe decreases special defense.

Poliwhirl foe using Hypnosis.

Minior ally is sleeping

Minior ally using Thunder Wave.

Poliwhirl foe is paralyzed

Poliwhirl foe using Bubble.

The target's Speed is lowering by one stage..

Critical hit!

Minior ally loses 18 hit points.

Minior ally faints.

Poliwag ally from the team Yellow enters the battle!

Poliwhirl foe using Hypnosis.

Poliwag ally is sleeping

Poliwag ally using Hypnosis.

Poliwhirl foe is sleeping

Poliwhirl foe using Facade.

Poliwag ally loses 6 hit points.

Poliwag ally using Hypnosis.

Poliwhirl foe is sleeping

Poliwag ally using Facade.

Poliwhirl foe loses 5 hit points.

Poliwhirl foe using Bubble.

The target's Speed is lowering by one stage..

Poliwag ally loses 3 hit points.

Poliwag ally using Facade.

Poliwhirl foe loses 6 hit points.

Poliwhirl foe faints.

Poliwrath foe from the team Blue enters the battle!

Poliwag ally using Facade.

Poliwrath foe loses 5 hit points.

Poliwrath foe using Bubble.

The target's Speed is lowering by one stage..

Poliwag ally loses 2 hit points.

Poliwag ally using Hypnosis.

Poliwrath foe is sleeping

Poliwrath foe using Poison Jab.

Poliwag ally loses 5 hit points.

Poliwag ally faints.

Team Yellow loses its last Pokemon.

The team Blue wins the battle!

Вывод:

Во время выполнения данной лабораторной работы я изучил основные принципы ООП, научился работать с методами, классами, модификаторами доступами и сторонними библиотеками.

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