

Факультет программной инженерии и

компьютерной техники

Лабораторная работа №2

**Основы объектно-ориентированного программирования**

Вариант №30082

Выполнила Громилова Мария Дмитриевна,

группа P3130

Преподаватель Блохина Елена Николаевна

г. Санкт-Петербург 2021 г.

**Оглавление:**

[Задание: 3](#_Toc87305882)

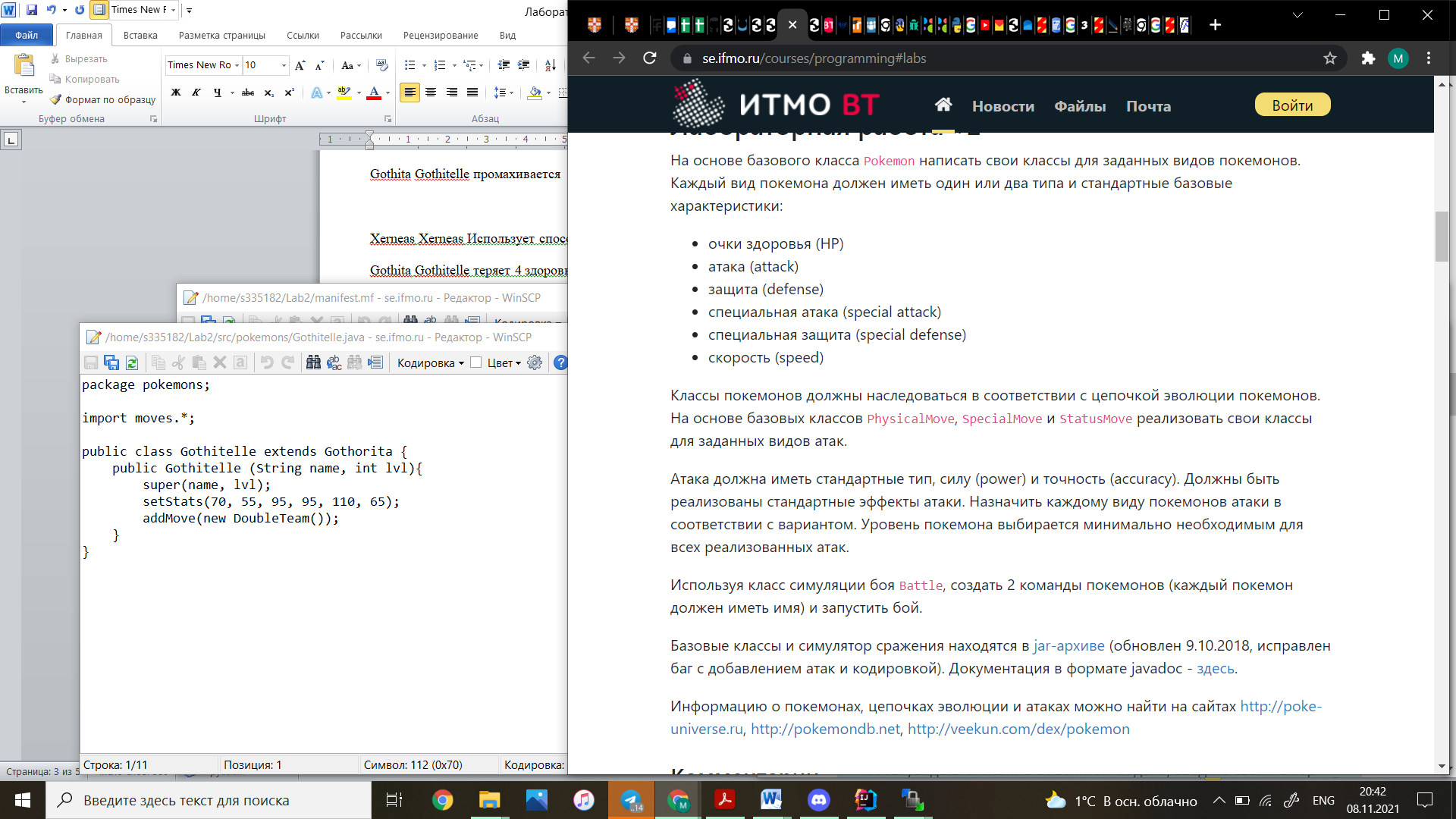
[Диаграмма классов реализованной объектной модели: 3](#_Toc87305884)

[Исходный код: 4](#_Toc87305885)

[Результаты работы программы: 12](#_Toc87305886)

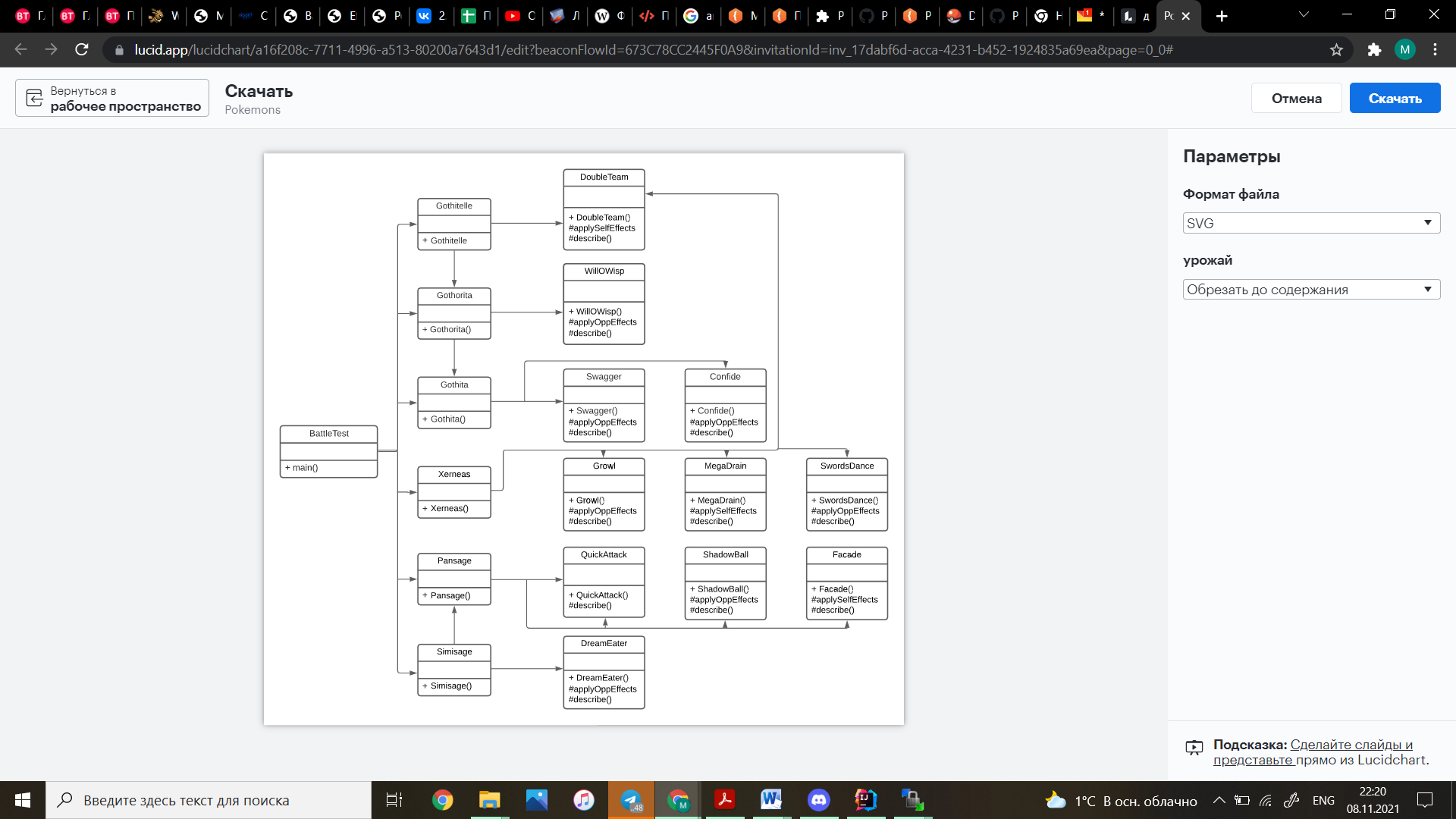
[Вывод: 15](#_Toc87305887)

# Задание:



# 

# Диаграмма классов реализованной объектной модели:



# Исходный код:

**BattleTest.java**

**import ru.ifmo.se.pokemon.\*;  
import pokemons.\*;  
  
class BattleTest {  
 public static void main(String[] args){  
 Battle b = new Battle();  
 Pokemon p1 = new Xerneas("Xerneas", 2);  
 Pokemon p2 = new Simisage("Simisage", 3);  
 Pokemon p3 = new Pansage("Pansage", 1);  
 Pokemon p4 = new Gothita("Gothitelle", 3);  
 Pokemon p5 = new Gothitelle("Gothita", 4);  
 Pokemon p6 = new Gothorita("Gothorita", 1);  
  
 b.addAlly(p1);  
 b.addAlly(p2);  
 b.addAlly(p3);  
 b.addFoe(p4);  
 b.addFoe(p5);  
 b.addFoe(p6);  
  
 b.go();  
 }  
}**

**Xerneas.java**

**package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Xerneas extends Pokemon {  
 public Xerneas(String name, int lvl){  
 super(name, lvl);  
 setStats(126,131, 95, 131, 98, 99);  
 setType(Type.*FAIRY*);  
 setMove(new DoubleTeam(), new Growl(), new MegaDrain(), new SwordsDance());  
 }  
}**

**Simisage.java**

**package pokemons;  
  
import moves.\*;  
  
public class Simisage extends Pansage{  
 public Simisage (String name, int lvl){  
 super(name, lvl);  
 setStats(75, 98, 63, 98, 63, 101);  
 addMove(new DreamEater());  
 }  
}**

**Pansage.java**

**package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Pansage extends Pokemon{  
 public Pansage(String name, int lvl){  
 super(name, lvl);  
 setStats(50, 53, 48, 53, 48, 64);  
 setType(Type.*GRASS*);  
 setMove(new QuickAttack(), new ShadowBall(), new Facade());  
 }  
}**

**Gothorita.java**

**package pokemons;  
  
import moves.\*;  
  
public class Gothorita extends Gothita {  
 public Gothorita (String name, int lvl){  
 super(name, lvl);  
 setStats(60, 45, 70, 75, 85, 55);  
 addMove(new WillOWisp());  
 }  
}**

**Gothitelle.java**

**package pokemons;  
  
import moves.\*;  
  
public class Gothitelle extends Gothorita {  
 public Gothitelle (String name, int lvl){  
 super(name, lvl);  
 setStats(70, 55, 95, 95, 110, 65);  
 addMove(new DoubleTeam());  
 }  
}**

**Gothita.java**

**package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Gothita extends Pokemon{  
 public Gothita(String name, int lvl){  
 super(name, lvl);  
 setStats(45, 30, 50, 55, 65, 45);  
 setType(Type.*PSYCHIC*);  
 setMove(new Swagger(), new Confide());  
 }  
}**

**Confide.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Confide extends StatusMove {  
 *// Устанавливаем базовые характеристики* public Confide(){  
 super(Type.*NORMAL*, 0, 0);  
 }  
  
 *// Снижает специальную аттаку на 1 ступень* @Override  
 protected void applyOppEffects(Pokemon p){  
 Effect e = new Effect().stat(Stat.*ATTACK*, -1);  
 p.addEffect(e);  
 }  
 *// Сообщение об использовании* @Override  
 protected String describe(){  
 return "Использует способность Confide";  
 }  
}**

**DoubleTeam.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DoubleTeam extends StatusMove {  
 *// Устанавливаем базовые характеристики* public DoubleTeam(){  
 super(Type.*NORMAL*,0.0,0.0);  
 }  
 *// Повышает свое уклонение на 1 ступень* @Override protected void applySelfEffects(Pokemon p){  
 Effect e = new Effect().stat(Stat.*EVASION*, 1);  
 p.addEffect(e);  
 }  
 *// Сообщение об использовании* @Override  
 protected String describe(){  
 return "Использует способность DoubleTeam";  
 }  
}**

**DreamEater.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DreamEater extends SpecialMove{  
 public DreamEater(){  
 super(Type.*PSYCHIC*, 100.0, 1.0);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 Effect e = new Effect().condition(Status.*SLEEP*);  
 p.addEffect(e);  
 }  
  
 *// Сообщение об использовании* @Override  
 protected String describe(){  
 return "Использует способность DreamEater";  
 }  
}**

**Façade.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Facade extends PhysicalMove {  
 *// Устанавливаем базовые характеристики* public Facade() {  
 super(Type.*NORMAL*, 70.0, 1.0);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 Effect e = new Effect().stat(Stat.*ATTACK*, 2);  
 p.addEffect(e);  
 }  
  
  
 @Override  
 protected String describe(){  
 return "Использует способность ShadowBall";  
 }  
}**

**Growl.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Growl extends StatusMove{  
 *// Устанавливаем базовые характеристики* public Growl(){  
 super(Type.*NORMAL*, 0.0, 1.0);  
 }  
 *// Снижает атаку цели на 1* @Override  
 protected void applyOppEffects(Pokemon p){  
 Effect e = new Effect().stat(Stat.*ATTACK*, -1);  
 p.addEffect(e);  
 }  
 *// Сообщение об использовании* @Override  
 protected String describe(){  
 return "Использует способность Growl";  
 }  
  
}**

**MegaDrain**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class MegaDrain extends SpecialMove {  
 *// Устанавливаем базовые характеристики* public MegaDrain(){  
 super(Type.*GRASS*, 40.0, 1.0);  
 }  
  
  
 @Override  
 protected void applySelfEffects(Pokemon p){  
 Effect e = new Effect().chance(0.2).stat(Stat.*HP*, 2);  
 p.addEffect(e);  
 }  
 *// Сообщение об использовании* @Override  
 protected String describe(){  
 return "Использует способность MegaDrain";  
 }  
  
}**

**QuickAttack.java**

**package moves;**

**import ru.ifmo.se.pokemon.\*;  
  
public class QuickAttack extends PhysicalMove {  
 *// Устанавливаем базовые характеристики* public QuickAttack(){  
 super(Type.*FIRE*, 40.0, 1.0, 1, 1);  
 }  
 *// Сообщение об использовании* @Override  
 protected String describe(){  
 return "Использует способность QuickAttack";  
 }  
  
}**

**ShadowBall.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class ShadowBall extends SpecialMove{  
 *// Устанавливаем базовые характеристики* public ShadowBall() {  
 super(Type.*GHOST*, 80.0, 1.0);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 Effect e = new Effect().chance(0.2).stat(Stat.*DEFENSE*, -2);  
 p.addEffect(e);  
 }  
  
 @Override  
 protected String describe(){  
 return "Использует способность ShadowBall";  
 }  
}**

**Swagger.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Swagger extends StatusMove {  
 *// Устанавливаем базовые характеристики* public Swagger(){  
 super(Type.*NORMAL*,0.0,0.85);  
 }  
 *// Сбивает цель с току и повышает атаку на 2 уровня* @Override protected void applyOppEffects(Pokemon p) {  
 Effect e = new Effect().stat(Stat.*ATTACK*, 2);  
 e.*confuse*(p);  
 p.addEffect(e);  
 }  
 *// Сообщение об использовании* @Override  
 protected String describe(){  
 return "Использует способность Swagger";  
 }  
}**

**SwordsDance.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class SwordsDance extends StatusMove{  
 *// Устанавливаем базовые характеристики* public SwordsDance() {  
 super(Type.*NORMAL*, 0.0, 1.0);  
 }  
 *// Увеличивает атаку покемона на 2 уровня* @Override protected void applyOppEffects(Pokemon p) {  
 Effect e = new Effect().stat(Stat.*ATTACK*, 2);  
 p.addEffect(e);  
 }  
 @Override  
 protected String describe(){  
 return "Использует способность SwordsDance";  
 }  
}**

**WillOWisp.java**

**package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class WillOWisp extends StatusMove {  
 *// Устанавливаем базовые характеристики* public WillOWisp(){  
 super(Type.*FIRE*, 10.0, 0.85);  
 }  
 *// При попадении противник сгорает* @Override  
 protected void applyOppEffects(Pokemon p){  
 Effect e = new Effect().stat(Stat.*ATTACK*, -3).condition(Status.*BURN*);  
 p.addEffect(e);  
 }  
 *// Сообщение об использовании* @Override  
 protected String describe(){  
 return "Использует способность Will-O-Wisp";  
 }  
  
}**

# Результаты работы программы:

Xerneas Xerneas из команды синих вступает в бой!

Gothita Gothitelle из команды желтых вступает в бой!

Xerneas Xerneas Использует способность SwordsDance.

Gothita Gothitelle промахивается

Xerneas Xerneas Использует способность SwordsDance.

Gothita Gothitelle промахивается

Xerneas Xerneas Использует способность Growl.

Gothita Gothitelle Использует способность Swagger.

Xerneas Xerneas Использует способность SwordsDance.

Gothita Gothitelle Использует способность Swagger.

Xerneas Xerneas промахивается

Gothita Gothitelle Использует способность Swagger.

Xerneas Xerneas Использует способность Growl.

Gothita Gothitelle Использует способность Swagger.

Xerneas Xerneas промахивается

Gothita Gothitelle промахивается

Xerneas Xerneas Использует способность MegaDrain.

Gothita Gothitelle теряет 6 здоровья.

Gothita Gothitelle Использует способность Swagger.

Xerneas Xerneas теряет 2 здоровья.

Xerneas Xerneas Использует способность SwordsDance.

Gothita Gothitelle промахивается

Xerneas Xerneas теряет 2 здоровья.

Xerneas Xerneas промахивается

Gothita Gothitelle промахивается

Xerneas Xerneas теряет 2 здоровья.

Xerneas Xerneas промахивается

Gothita Gothitelle Использует способность Swagger.

Xerneas Xerneas теряет 2 здоровья.

Xerneas Xerneas растерянно попадает по себе.

Xerneas Xerneas теряет 8 здоровья.

Gothita Gothitelle промахивается

Xerneas Xerneas теряет 2 здоровья.

Simisage Simisage из команды синих вступает в бой!

Simisage Simisage Использует способность QuickAttack.

Gothita Gothitelle теряет 6 здоровья.

Gothita Gothitelle Использует способность Swagger.

Simisage Simisage Использует способность QuickAttack.

Gothita Gothitelle теряет 5 здоровья.

Gothita Gothitelle теряет сознание.

Gothitelle Gothita из команды желтых вступает в бой!

Simisage Simisage Использует способность ShadowBall.

Критический удар!

Gothitelle Gothita теряет 26 здоровья.

Gothitelle Gothita теряет сознание.

Gothorita Gothorita из команды желтых вступает в бой!

Simisage Simisage растерянно попадает по себе.

Simisage Simisage теряет 8 здоровья.

Gothorita Gothorita Использует способность Will-O-Wisp.

Simisage Simisage воспламеняется

Simisage Simisage Использует способность DreamEater.

Gothorita Gothorita теряет 4 здоровья.

Gothorita Gothorita засыпает

Gothorita Gothorita Использует способность Will-O-Wisp.

Simisage Simisage Использует способность QuickAttack.

Gothorita Gothorita теряет 8 здоровья.

Gothorita Gothorita Использует способность Will-O-Wisp.

Simisage Simisage Использует способность ShadowBall.

Gothorita Gothorita теряет 6 здоровья.

Gothorita Gothorita теряет сознание.

В команде желтых не осталось покемонов.

Команда синих побеждает в этом бою!

# Вывод:

В ходе лабораторной работы я изучила основные принципы ООП и научилась применять их на практике.