Санкт-Петербургский национально исследовательский университет

информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники



**Лабораторная работа № 2**

Вариант № 3112220

Выполнил: Балтабаев Дамир Темиржанович

Группа: P3112

Преподаватели: Исаев Александр Сергеевич

Каюков Иван Алексеевич

г. Санкт-Петербург

2020

**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)

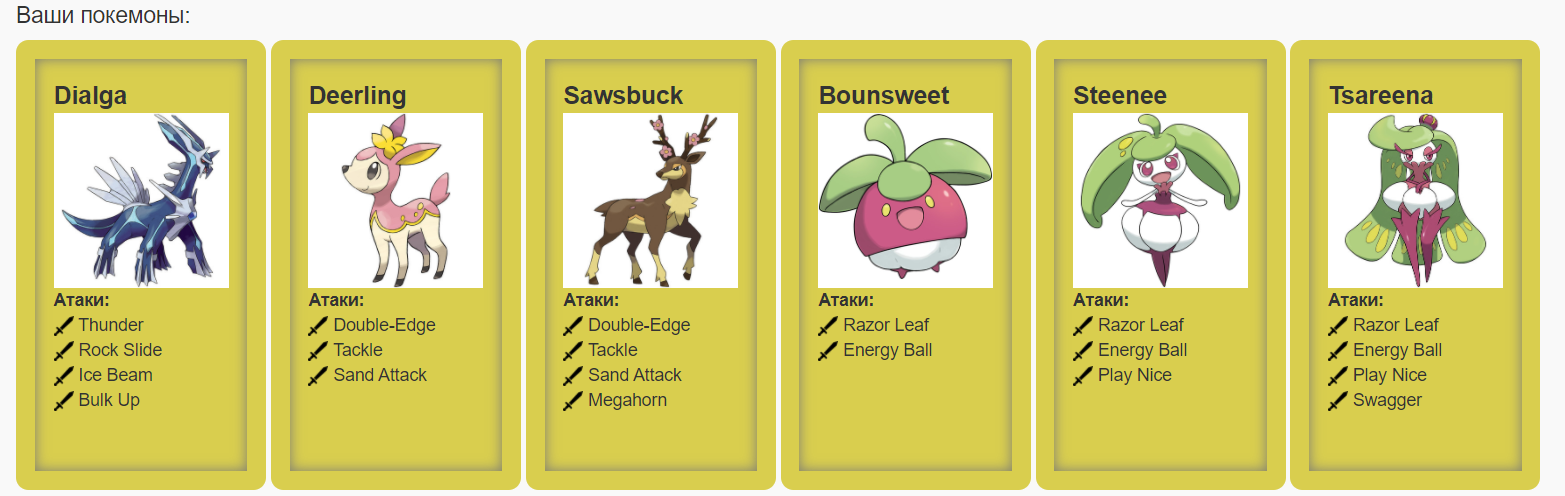
Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов PhysicalMove, SpecialMove и StatusMove реализовать свои классы для заданных видов атак.

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.

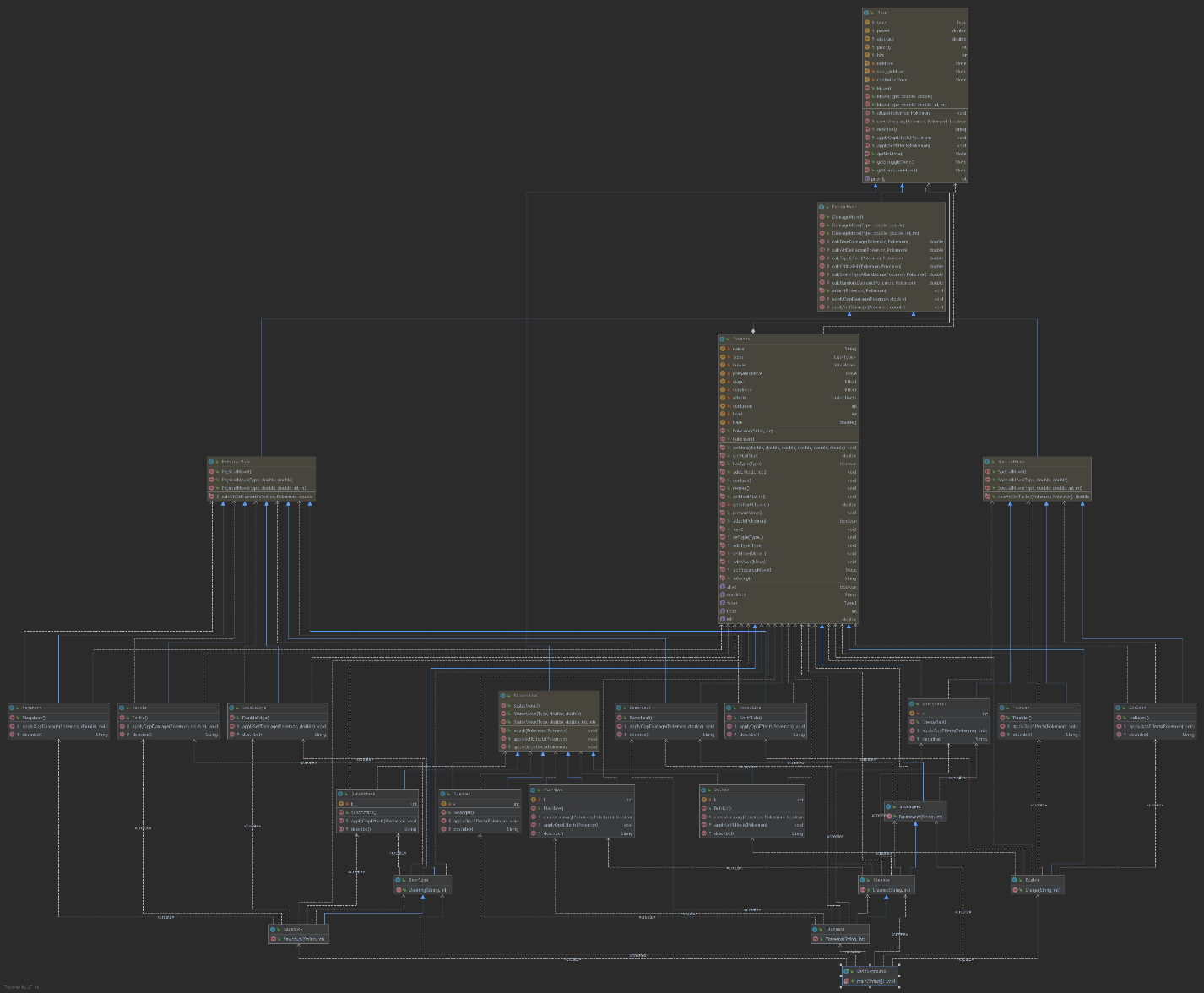
Используя класс симуляции боя Battle, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

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://poke-universe.ru/), [http://pokemondb.net](http://pokemondb.net/),<http://veekun.com/dex/pokemon>



**Диаграмма классов**



**Код**

[**https://github.com/damir2407/Lab2.git**](https://github.com/damir2407/Lab2.git)

Battleground.java

package MyPokemons;  
  
import ru.ifmo.se.pokemon.Battle;  
  
public class Battleground {  
  
 public static void main(String[] args) {  
 Battle b = new Battle();  
 b.addAlly(new Dialga("Макгрегор", 7));  
 b.addFoe(new Deerling("Нурмагомедов", 7));  
 b.addAlly(new Sawsbuck("Тайсон", 6));  
 b.addFoe(new Bounsweet("Али", 6));  
 b.addAlly(new Steenee("Мейвезер", 8));  
 b.addFoe(new Tsareena("Пакьяо", 8));  
 b.go();  
 }  
}

Bounsweet.java

package MyPokemons;  
  
import PokemonsAttacks.EnergyBall;  
import PokemonsAttacks.RazorLeaf;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Bounsweet extends Pokemon {  
 public Bounsweet(String name,int lvl){  
 super(name,lvl);  
 setType(Type.*GRASS*);  
 setStats(42,30,38,30,38,32);  
 setMove(new RazorLeaf(),new EnergyBall());  
 }  
}

Deerling.java

package MyPokemons;  
  
import PokemonsAttacks.DoubleEdge;  
import PokemonsAttacks.SandAttack;  
import PokemonsAttacks.Tackle;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Deerling extends Pokemon {  
 public Deerling(String name,int lvl){  
 super(name,lvl);  
 setType(Type.*NORMAL*,Type.*GRASS*);  
 setStats(60,60,50,40,50,75);  
 setMove(new DoubleEdge(),new Tackle(),new SandAttack());  
 }  
}

Dialga.java

package MyPokemons;  
  
import PokemonsAttacks.BulkUp;  
import PokemonsAttacks.IceBeam;  
import PokemonsAttacks.RockSlide;  
import PokemonsAttacks.Thunder;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Dialga extends Pokemon {  
 public Dialga(String name, int lvl){  
 super(name,lvl);  
 setStats(100,120,120,150,100,90);  
 setType(Type.*STEEL*,Type.*DRAGON*);  
 setMove(new Thunder(),new BulkUp(),new IceBeam(),new RockSlide());  
 }  
}

Sawsbuck.java

package MyPokemons;  
  
import PokemonsAttacks.DoubleEdge;  
import PokemonsAttacks.Megahorn;  
import PokemonsAttacks.SandAttack;  
import PokemonsAttacks.Tackle;  
import ru.ifmo.se.pokemon.Type;  
  
public class Sawsbuck extends Deerling{  
 public Sawsbuck(String name,int lvl){  
 super(name,lvl);  
 setType(Type.*NORMAL*,Type.*GRASS*);  
 setStats(80,100,70,60,70,95);  
 setMove(new DoubleEdge(),new Tackle(),new SandAttack(),new Megahorn());  
 }  
}

Steenee.java

package MyPokemons;  
  
import PokemonsAttacks.EnergyBall;  
import PokemonsAttacks.PlayNice;  
import PokemonsAttacks.RazorLeaf;  
import ru.ifmo.se.pokemon.Type;  
  
public class Steenee extends Bounsweet {  
 public Steenee(String name, int lvl) {  
 super(name, lvl);  
 setStats(52, 40, 48, 40, 48, 62);  
 setType(Type.*GRASS*);  
 setMove(new RazorLeaf(),new EnergyBall(),new PlayNice());  
 }  
  
}

Tsareena.java

package MyPokemons;  
  
import PokemonsAttacks.EnergyBall;  
import PokemonsAttacks.PlayNice;  
import PokemonsAttacks.RazorLeaf;  
import PokemonsAttacks.Swagger;  
import ru.ifmo.se.pokemon.Type;  
  
public class Tsareena extends Steenee {  
 public Tsareena(String name,int lvl){  
 super(name,lvl);  
 setStats(72,120,98,50,98,72);  
 setType(Type.*GRASS*);  
 setMove(new RazorLeaf(),new EnergyBall(),new PlayNice(),new Swagger());  
 }  
}

BulkUp.java

package PokemonsAttacks;  
  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.StatusMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class BulkUp extends StatusMove {  
 private int k = 0;  
  
 public BulkUp() {  
 super(Type.*FIGHTING*, 0, 0);  
 }  
  
 @Override  
 protected boolean checkAccuracy(Pokemon att,  
 Pokemon def) {  
 return true;  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 k = k + 1;  
 if (k < 7) {  
 p.setMod(Stat.*ATTACK*, +1);  
 p.setMod(Stat.*DEFENSE*, +1);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 if (k < 6) {  
 return "использует способность Bulk Up";  
 } else {  
 return "пытается использовать способность Bulk Up. Неудача";  
 }  
  
 }  
}

DoubleEdge.java

package PokemonsAttacks;  
  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
  
public class DoubleEdge extends PhysicalMove {  
 public DoubleEdge() {  
 super(Type.*NORMAL*, 120, 100);  
 }  
  
 @Override  
 protected void applySelfDamage(Pokemon att,  
 double damage) {  
 att.setMod(Stat.*HP*, (int) Math.*round*(damage \* 1 / 3d));  
 }  
  
 @Override  
 protected String describe() {  
 return "использует способность Double Edge, нанося себе треть урона";  
 }  
}

EnergyBall.java

package PokemonsAttacks;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
  
public class EnergyBall extends SpecialMove {  
 private int k;  
  
 public EnergyBall() {  
 super(Type.*GRASS*, 90, 100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 k = k + 1;  
 if ((Math.*random*() <= 0.1) && (k < 7)) {  
 p.setMod(Stat.*SPECIAL\_DEFENSE*, -1);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 if (k < 6) {  
 return "использует способность Energy Ball, с шансом снизить специальную защиту соперника на 1";  
 } else return "пытается использовать способность Energy Ball. Неудача";  
  
 }  
}

IceBeam.java

package PokemonsAttacks;  
  
import ru.ifmo.se.pokemon.Effect;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Type;  
  
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 "использует способность Ice Beam, с шансом 10% заморозить соперника";  
 }  
}

Megahorn.java

package PokemonsAttacks;  
  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
  
public class Megahorn extends PhysicalMove {  
 public Megahorn() {  
 super(Type.*BUG*, 120, 85);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def,  
 double damage) {  
 def.setMod(Stat.*HP*, (int) Math.*round*(damage));  
 }  
  
 @Override  
 protected String describe() {  
 return "использует способность Megahorn";  
 }  
  
}

PlayNice.java

package PokemonsAttacks;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.StatusMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class PlayNice extends StatusMove {  
 private int k;  
  
 public PlayNice() {  
 super(Type.*NORMAL*, 0, 0);  
 }  
  
 @Override  
 protected boolean checkAccuracy(Pokemon, Pokemon pokemon1) {  
 return true;  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 k = k + 1;  
 if (k < 7) {  
 pokemon.setMod(Stat.*ATTACK*, -1);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 if (k < 6) {  
 return "использует способность Play Nice";  
 } else return "пытается использовать способность Play Nice. Неудача";  
  
  
 }  
}

RazorLeaf.java

package PokemonsAttacks;  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
public class RazorLeaf extends PhysicalMove {  
 public RazorLeaf() {  
 super(Type.*GRASS*, 55, 95);  
 }  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage) {  
 if (Math.*random*() <= 0.125) {  
 System.*out*.println("Критический удар!");  
 def.setMod(Stat.*HP*, (int) Math.*round*(damage) \* 2);  
 } else {  
 def.setMod(Stat.*HP*, (int) Math.*round*(damage) \* 1);  
 }  
 }  
 @Override  
 protected String describe() {  
 return "использует способность Razor Leaf с повышенным шансом на критический удар";  
 }  
}

RockSlide.java

package PokemonsAttacks;  
  
import ru.ifmo.se.pokemon.Effect;  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class RockSlide extends PhysicalMove {  
 public RockSlide() {  
 super(Type.*ROCK*, 75, 90);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.*random*() <= 0.3) {  
 Effect.*flinch*(p);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует способность Rock Slide, с шансом 30% заставить вздрогнуть соперника";  
 }  
}

SandAttack.java

package PokemonsAttacks;  
  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.StatusMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class SandAttack extends StatusMove {  
 private int k;  
  
 public SandAttack() {  
 super(Type.*GROUND*, 0, 100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 k = k + 1;  
 if (k < 7) {  
 p.setMod(Stat.*ACCURACY*, -1);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 if (k < 6) {  
 return "использует способность Sand Attack";  
 } else {  
 return "пытается использовать способность Sand Attack. Неудача";  
 }  
  
 }  
}

Swagger.java

package PokemonsAttacks;  
  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Swagger extends StatusMove {  
 private int k;  
  
 public Swagger() {  
 super(Type.*NORMAL*, 0, 85);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 k = k + 1;  
 if (k < 7) {  
 Effect.*confuse*(p);  
 p.setMod(Stat.*ATTACK*, +2);  
 p.addEffect(new Effect().turns(4).chance(0.33).stat(Stat.*HP*, ((int) p.getStat(Stat.*ATTACK*))));  
 }  
 }  
  
  
 @Override  
 protected String describe() {  
 if (k < 6) {  
 return "использует способность Swagger";  
 } else return "пытается использовать способность Swagger. Неудача";  
 }  
}

Tackle.java

package PokemonsAttacks;  
  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
  
public class Tackle extends PhysicalMove {  
 public Tackle() {  
 super(Type.*NORMAL*, 40, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def,  
 double damage) {  
 def.setMod(Stat.*HP*, (int) Math.*round*(damage));  
 }  
  
 @Override  
 protected String describe() {  
 return "использует способность Tackle";  
 }  
}

Thunder.java

package PokemonsAttacks;  
  
import ru.ifmo.se.pokemon.Effect;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class Thunder extends SpecialMove {  
 public Thunder() {  
 super(Type.*ELECTRIC*, 110, 70);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if ((Math.*random*() <= 0.3) && (!p.hasType(Type.*ELECTRIC*))) {  
 Effect.*paralyze*(p);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует способность Thunder, с шансом 30% парализовать соперника";  
 }  
}

**Результат работы программы:**

Dialga Макгрегор из команды синих вступает в бой!

Deerling Нурмагомедов из команды белых вступает в бой!

Dialga Макгрегор использует способность Rock Slide, с шансом 30% заставить вздрогнуть соперника.

Deerling Нурмагомедов теряет 10 здоровья.

Deerling Нурмагомедов использует способность Tackle.

Dialga Макгрегор теряет 2 здоровья.

Dialga Макгрегор использует способность Ice Beam, с шансом 10% заморозить соперника.

Deerling Нурмагомедов теряет 26 здоровья.

Deerling Нурмагомедов теряет сознание.

Bounsweet Али из команды белых вступает в бой!

Dialga Макгрегор использует способность Ice Beam, с шансом 10% заморозить соперника.

Критический удар!

Bounsweet Али теряет 48 здоровья.

Bounsweet Али замерзает

Bounsweet Али теряет сознание.

Tsareena Пакьяо из команды белых вступает в бой!

Dialga Макгрегор использует способность Rock Slide, с шансом 30% заставить вздрогнуть соперника.

Tsareena Пакьяо теряет 5 здоровья.

Tsareena Пакьяо использует способность Swagger.

Dialga Макгрегор увеличивает атаку.

Dialga Макгрегор теряет 39 здоровья.

Sawsbuck Тайсон из команды синих вступает в бой!

Tsareena Пакьяо использует способность Swagger.

Sawsbuck Тайсон увеличивает атаку.

Sawsbuck Тайсон использует способность Tackle.

Tsareena Пакьяо теряет 11 здоровья.

Sawsbuck Тайсон теряет 29 здоровья.

Steenee Мейвезер из команды синих вступает в бой!

Tsareena Пакьяо использует способность Swagger.

Steenee Мейвезер увеличивает атаку.

Steenee Мейвезер использует способность Razor Leaf с повышенным шансом на критический удар.

Tsareena Пакьяо теряет 4 здоровья.

Steenee Мейвезер теряет 19 здоровья.

Tsareena Пакьяо использует способность Swagger.

Steenee Мейвезер увеличивает атаку.

Steenee Мейвезер использует способность Energy Ball, с шансом снизить специальную защиту соперника на 1.

Tsareena Пакьяо теряет 6 здоровья.

Steenee Мейвезер теряет 19 здоровья.

Steenee Мейвезер теряет 25 здоровья.

В команде синих не осталось покемонов.

Команда белых побеждает в этом бою!

**Вывод**

В ходе выполнения данной лабораторной работы я познакомился с основами объектно-ориентированного программирования и узнал основные его принципы, такие как наследование, полиморфизм и инкапсуляция.