МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине

«ПРОГРАММИРОВАНИЕ»

Вариант № 153235

***Выполнил:***

Студент группы P3115

Собитов Анвархон Акмалович

***Преподаватель:***

Сорокин Роман Борисович

Санкт-Петербург, 2022

# **Текст 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.

# 

# ***Покемоны:***



## **Код программы:**

***Атаки:***

package myattack;  
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 AerialAce extends PhysicalMove {  
 public AerialAce(double pow, double acc){  
 super(Type.*FLYING*, pow, acc);  
 }  
 @Override  
 public void applySelfEffects (Pokemon p){  
 p.setMod(Stat.*SPEED*,-1);  
 }  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
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 Discharge extends SpecialMove {  
 public Discharge(double pow, double acc){  
 super(Type.*ELECTRIC*,pow, acc);  
  
 }  
 @Override  
 public void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*ATTACK*,+2);  
 }@Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class DoubleKick extends PhysicalMove {  
 public DoubleKick(double pow , double acc){  
 super(Type.*FIGHTING*,pow ,acc);  
 }  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage){  
 super.applyOppDamage(def,damage);  
 super.applyOppDamage(def,damage);

}  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.\*;  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Facade extends PhysicalMove {  
 public Facade(double pow, double acc){  
 super(Type.*NORMAL*, pow, acc);  
 }  
 private boolean flag;  
 @Override  
 public void applyOppDamage(Pokemon poc, double damage){  
 Status cond = poc.getCondition();  
 flag = true;  
 if (cond.equals(Status.*POISON*) || cond.equals(Status.*BURN*) || cond.equals(Status.*PARALYZE*)) {  
 poc.setMod(Stat.*HP*, -2\*(int)Math.*round*(damage));  
 }  
 }  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class PoisonTail extends PhysicalMove {  
 public PoisonTail(double pow, double acc){  
 super(Type.*POISON*, pow, acc);  
 }  
  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class SludgeBomb extends SpecialMove {  
 public SludgeBomb(double pow, double acc){  
 super(Type.*POISON*, pow, acc);

}  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class SludgeWave extends SpecialMove {  
 public SludgeWave(double pow, double acc){  
 super(Type.*POISON*, pow, acc);  
 }  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.\*;  
  
public class Swagger extends StatusMove {  
 public Swagger (double pow, double acc){  
 super(Type.*NORMAL*, pow, acc);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 super.applyOppEffects(p);  
 Effect.*confuse*(p);  
 }  
 @Override  
 protected void applySelfEffects(Pokemon p){  
 super.applySelfEffects(p);  
 Effect e = new Effect().stat(Stat.*ATTACK*, 2);  
 p.addEffect(e);  
 }  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.StatusMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class TailWhip extends StatusMove {  
 public TailWhip(double pow, double acc){  
 super(Type.*NORMAL*, pow, acc);  
 }  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.Effect;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.StatusMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class ThunderWave extends StatusMove {  
 public ThunderWave(double pow, double acc){  
 super(Type.*ELECTRIC*, pow, acc);  
 }  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

package myattack;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Type;  
public class Twister extends SpecialMove {  
 public Twister(double pow, double acc){  
 super(Type.*DRAGON*, pow, acc);  
 }  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage){  
 super.applyOppDamage(def,damage);  
 }  
  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];

}

}

package myattack;  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class XScissor extends PhysicalMove {  
 public XScissor(double pow, double acc){  
 super(Type.*BUG*, pow, acc);

}  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage){  
 super.applyOppDamage(def,damage);  
 }  
  
 @Override  
 protected String describe(){  
 String[] asd = this.getClass().toString().split("\\.");  
 return "does " + asd[asd.length-1];  
 }  
}

***Покемоны:***

import myattack.PoisonTail;  
import myattack.SludgeBomb;  
import myattack.SludgeWave;  
import myattack.Twister;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Dragalge extends Pokemon{  
 public Dragalge(String name, int level){  
 super(name,level);  
  
 super.setType(Type.*POISON*, Type.*DRAGON*);  
 super.setStats(65,75,90,97,123,44);  
  
 SludgeWave sludgeWave = new SludgeWave(95,100);  
 PoisonTail poisonTail = new PoisonTail(50,100);  
 SludgeBomb sludgeBomb = new SludgeBomb(90,100);  
 Twister twister = new Twister(40,100);  
  
 super.setMove(sludgeWave, poisonTail, sludgeBomb, twister);  
 }  
}

import myattack.Facade;  
import myattack.TailWhip;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Pichu extends Pokemon {  
 public Pichu(String name, int level){  
 super(name, level);  
  
 super.setType(Type.*ELECTRIC*);  
 super.setStats(20,40,15,35,35,60);  
  
 Facade facade = new Facade(70,100);  
 TailWhip tailWhip = new TailWhip(0,100);  
 super.setMove(facade,tailWhip);  
 }  
}

import myattack.Discharge;  
import myattack.Facade;  
import myattack.TailWhip;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Pikachu extends Pokemon {  
 public Pikachu(String name, int level){  
 super(name,level);  
  
 super.setType(Type.*ELECTRIC*);  
 super.setStats(35,55,40,50,50,90);  
  
 Facade facade = new Facade(70,100);  
 TailWhip tailWhip = new TailWhip(0,100);  
 Discharge discharge = new Discharge(80,100);  
  
 super.setMove(facade, tailWhip, discharge);  
 }  
}

import myattack.Discharge;  
import myattack.Facade;  
import myattack.TailWhip;  
import myattack.ThunderWave;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
public class Raichu extends Pokemon {  
 public Raichu(String name, int level){  
 super(name,level);  
  
 super.setType(Type.*ELECTRIC*);  
 super.setStats(60,90,55,90,80,110);  
  
 Facade facade = new Facade(70,100);  
 TailWhip tailWhip = new TailWhip(0,100);  
 Discharge discharge = new Discharge(80,100);  
 ThunderWave thunderWave = new ThunderWave(100,70);  
  
 super.setMove(facade, tailWhip, discharge, thunderWave);  
 }  
}

import myattack.PoisonTail;  
import myattack.SludgeBomb;  
import myattack.SludgeWave;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Skrelp extends Pokemon {  
 public Skrelp(String name, int level){  
 super(name,level);  
  
 super.setType(Type.*POISON*, Type.*WATER*);  
 super.setStats(50,60,60,60,60,30);  
  
 SludgeWave sludgeWave = new SludgeWave(95,100);  
 PoisonTail poisonTail = new PoisonTail(50,100);  
 SludgeBomb sludgeBomb = new SludgeBomb(90,100);  
 super.setMove(sludgeWave, poisonTail, sludgeBomb);  
 }  
}

import myattack.AerialAce;  
import myattack.DoubleKick;  
import myattack.Swagger;  
import myattack.XScissor;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Virizion extends Pokemon {  
 public Virizion(String name, int level){  
 super(name,level);  
  
 super.setType(Type.*GRASS*, Type.*FIGHTING*);  
 super.setStats(91,90,72,90,129,108);  
  
 Swagger swagger = new Swagger(140,85);  
 XScissor xScissor = new XScissor(80,100);  
 DoubleKick doubleKick = new DoubleKick(30,100);  
 AerialAce aerialAce = new AerialAce(60,100);  
 super.setMove(swagger,xScissor, doubleKick, aerialAce);  
 }  
  
}

***main.java***

import ru.ifmo.se.pokemon.Battle;  
import ru.ifmo.se.pokemon.Pokemon;  
  
public class Main {  
 public static void main(String[] args) {  
 Battle b = new Battle();  
 Dragalge p1 = new Dragalge("Pok1", 1);  
 Virizion p2 = new Virizion("Pok2", 1);  
 Skrelp p3 = new Skrelp("Pok3", 1);  
  
 Pikachu p4 = new Pikachu("Pok4",1);  
 Pichu p5 = new Pichu("Pok5", 1);  
 Raichu p6 = new Raichu("Pok6",1);  
  
 b.addAlly(p1);  
 b.addAlly(p3);  
 b.addAlly(p5);  
  
 b.addFoe(p2);  
 b.addFoe(p4);  
 b.addFoe(p6);  
 b.go();  
 }  
}

*Результат работы программы:*

Dragalge Pok1 из команды желтых вступает в бой!

Virizion Pok2 из команды черных вступает в бой!

Virizion Pok2 does DoubleKick.

Dragalge Pok1 теряет 2 здоровья.

Dragalge Pok1 теряет 2 здоровья.

Dragalge Pok1 does Twister.

Virizion Pok2 теряет 6 здоровья.

Virizion Pok2 does DoubleKick.

Dragalge Pok1 теряет 3 здоровья.

Dragalge Pok1 теряет 3 здоровья.

Dragalge Pok1 does PoisonTail.

Virizion Pok2 теряет 14 здоровья.

Virizion Pok2 теряет сознание.

Pikachu Pok4 из команды черных вступает в бой!

Pikachu Pok4 does Discharge.

Dragalge Pok1 теряет 2 здоровья.

Pikachu Pok4 увеличивает атаку.

Dragalge Pok1 does SludgeBomb.

Pikachu Pok4 теряет 8 здоровья.

Pikachu Pok4 does Discharge.

Dragalge Pok1 теряет 3 здоровья.

Pikachu Pok4 увеличивает атаку.

Dragalge Pok1 теряет сознание.

Skrelp Pok3 из команды желтых вступает в бой!

Pikachu Pok4 does Discharge.

Skrelp Pok3 теряет 14 здоровья.

Pikachu Pok4 увеличивает атаку.

Skrelp Pok3 теряет сознание.

Pichu Pok5 из команды желтых вступает в бой!

Pikachu Pok4 does TailWhip.

Pichu Pok5 does Facade.

Pikachu Pok4 does Facade.

Pichu Pok5 does Facade.

Pikachu Pok4 does TailWhip.

Pichu Pok5 does Facade.

Pikachu Pok4 does TailWhip.

Pichu Pok5 does Facade.

Pikachu Pok4 does Discharge.

Pichu Pok5 теряет 4 здоровья.

Pikachu Pok4 увеличивает атаку.

Pichu Pok5 does Facade.

Pikachu Pok4 does TailWhip.

Pichu Pok5 does Facade.

Pikachu Pok4 does Facade.

Pichu Pok5 does TailWhip.

Pikachu Pok4 does Facade.

Pichu Pok5 does TailWhip.

Pikachu Pok4 does Discharge.

Pichu Pok5 теряет 4 здоровья.

Pikachu Pok4 увеличивает атаку.

Pichu Pok5 does TailWhip.

Pikachu Pok4 does Facade.

Pichu Pok5 does TailWhip.

Pikachu Pok4 does Discharge.

Pichu Pok5 теряет 4 здоровья.

Pikachu Pok4 увеличивает атаку.

Pichu Pok5 теряет сознание.

В команде желтых не осталось покемонов.

Команда черных побеждает в этом бою!

## Выводы:

В процессе выполнения лабораторной работы я получил навыки использования

объектно-ориентированного подхода программирования при использовании языка Java.

Научился работать с классами, конструкторами, полями и модификаторами доступа.