**ФПИиКТ**

**Рабочий протокол и отчет по**

**лабораторной работе №2**

**Вариант №311595**

Угур Байар

Группа: P3115

Преподаватель: Письмак Алексей

Санкт-Петербург

2020г.

# Текст задания

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.

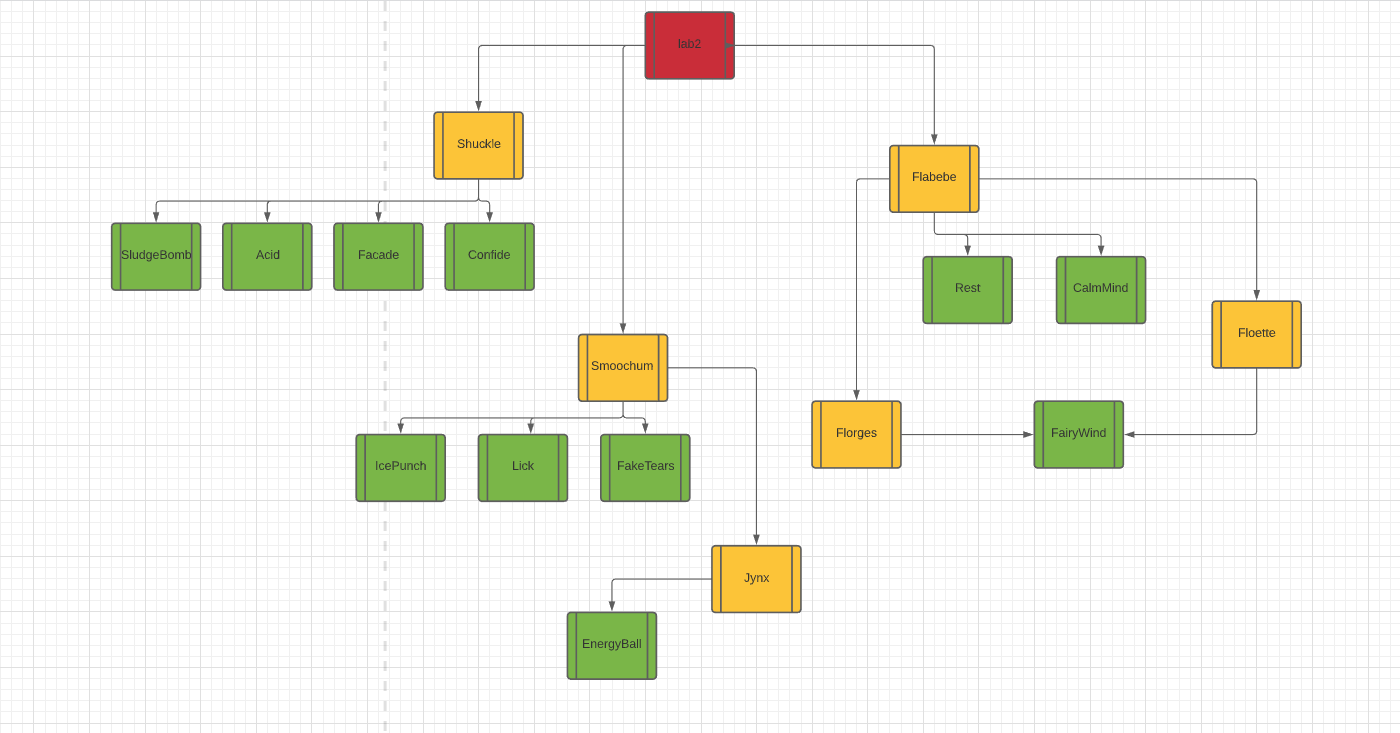
Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

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





# Диаграмма классов реализованной модели



Желтый – названия покемонов

Зеленый – названия атак

# Исходный код программы

# lab2.java

package com.company;  
  
import ru.ifmo.se.pokemon.Battle;  
  
public class Lab2{  
 public static void main(String args[]){  
 Battle b = new Battle();  
 b.addAlly(new Chansey("Чхансей", 1));  
 b.addAlly(new Blissey("Блиссей", 2));  
 b.addAlly(new Happiny("Хаппинй", 3));  
 b.addFoe(new TapuFini("Тапу Фини", 1));  
 b.addFoe(new Eevee ("Еевее", 2));  
 b.addFoe(new Umbreon("Умбрейон", 3));  
 b.go();  
 }//формируем бой  
}

# TapuFini.java

package com.company;  
import ru.ifmo.se.pokemon.\*;  
  
class TapuFini extends Pokemon{  
 public TapuFini(String name, int level){  
 super(name, level);  
 this.setStats(70,75,115,95,130,85);  
 this.setType(Type.*FAIRY*);  
 Move[] Attack = new Move[4];  
 Attack [0] = new BulkUp();  
 Attack [1] = new RockTomb();  
 Attack [2] = new ZenHeadbutt();  
 Attack [3] = new SmartStrike();  
 this.setMove(Attack);  
 }  
}  
  
  
class BulkUp extends StatusMove{  
 public BulkUp(){}  
 Effect BulkUp = new Effect().chance(0.3).turns(1).stat(Stat.*ATTACK*, 1).stat(Stat.*DEFENSE*,1);  
 protected void applySelfEffects(Pokemon att){att.addEffect(BulkUp);}  
 protected String describe(){return ("использует BulkUp");}  
}  
  
class RockTomb extends PhysicalMove{  
 public RockTomb(){super(Type.*ROCK*,60,95);}  
 Effect RockTomb = new Effect().chance(0.3).turns(1).stat(Stat.*SPEED*, -1);  
 protected void applyOppEffects(Pokemon att){att.addEffect(RockTomb);}  
 protected String describe(){return ("использует RockTomb");}  
}  
  
  
class SmartStrike extends PhysicalMove{  
 public SmartStrike(){super(Type.*STEEL*,70,0);}  
 protected String describe(){return ("использует SmartStrike");}  
}  
  
class ZenHeadbutt extends PhysicalMove{  
 public ZenHeadbutt(){super(Type.*PSYCHIC*,80,90);}  
 Effect ZenHeadbutt = new Effect().chance(0.2).turns(1);  
 protected void applyOppEffects(Pokemon att){att.addEffect(ZenHeadbutt);}  
 protected String describe(){return ("использует ZenHeadbutt");}  
}

Umbreon.java

package com.company;  
import ru.ifmo.se.pokemon.\*;  
  
class Umbreon extends Eevee{  
 public Umbreon(String name, int level){  
 super(name, level);  
 this.setStats(95,65,110,60,130,65);  
 this.setType(Type.*DARK*);  
 Move[] Attack = new Move[4];  
 Attack [0] = new DoubleTeam();  
 Attack [1] = new WorkUp();  
 Attack [2] = new Facade();  
 Attack [3] = new DarkPulse();  
 this.setMove(Attack);  
 }  
}  
  
class DarkPulse extends SpecialMove{  
 public DarkPulse(){super(Type.*DARK*,80D,100);}  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.*random*() < 0.2) {  
 Effect.*flinch*(p);  
 }  
 }  
 protected String describe(){return ("использует Acid");}  
}

# Eevee.java

package com.company;  
import ru.ifmo.se.pokemon.\*;  
  
class Eevee extends Pokemon{  
 public Eevee(String name, int level){  
 super(name, level);  
 this.setStats(55,55,50,45,65,55);  
 this.setType(Type.*NORMAL*);  
 Move[] Attack = new Move[3];  
 Attack [0] = new DoubleTeam();  
 Attack [1] = new WorkUp();  
 Attack [2] = new Facade();  
 this.setMove(Attack);  
 }  
}  
  
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 "использует Double Team";  
 }  
}  
class WorkUp extends StatusMove {  
 public WorkUp() {  
 }  
 protected String describe() {  
 return ("использует WorkUp");  
 }  
}  
class Facade extends PhysicalMove {  
 public Facade() {  
 super(Type.*NORMAL*, 70, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage) {  
 Status p\_stat = def.getCondition();  
 if (p\_stat.equals(Status.*BURN*) || p\_stat.equals(Status.*POISON*) || p\_stat.equals(Status.*PARALYZE*)) {  
 super.applyOppDamage(def, damage\*2);  
 }  
 else {  
 super.applyOppDamage(def, damage);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Facade";  
 }  
}

# Happiny.java

package com.company;  
import ru.ifmo.se.pokemon.\*;  
  
class Happiny extends Pokemon{  
 public Happiny(String name, int level){  
 super(name, level);  
 this.setStats(100,5,5,15,65,30);  
 this.setType(Type.*NORMAL*);  
 Move[] Attack = new Move[2];  
 Attack [0] = new Swagger();  
 Attack [1] = new ShadowBall();  
 this.setMove(Attack);  
 }  
}  
  
  
class Swagger extends StatusMove{  
 public Swagger(){super(Type.*NORMAL*,0,85);}  
 Effect Swagger = new Effect().chance(0.3).turns(1).stat(Stat.*ATTACK*, 2);  
 protected void applySelfEffects(Pokemon def){def.addEffect(Swagger);}  
 protected void applyOppEffects(Pokemon att){Effect.*confuse*(att);}  
 protected String describe(){return ("использует Swagger");}  
  
}  
  
class ShadowBall extends SpecialMove {  
 public ShadowBall() {  
 super(Type.*GHOST*, 80D, 100);  
 }  
  
 Effect ShadowBall = new Effect().chance(0.2).turns(1).stat(Stat.*SPECIAL\_DEFENSE*, -1);  
  
 protected void applyOppEffects(Pokemon p) {  
 p.addEffect(ShadowBall);  
 }  
  
 protected String describe() {  
 return ("использует ShadowBall");  
 }

# Blissey.java

package com.company;  
import ru.ifmo.se.pokemon.\*;  
  
class Blissey extends Chansey{  
 public Blissey(String name, int level){  
 super(name, level);  
 this.setStats(255,30,15,85,65,65);  
 this.setType(Type.*NORMAL*);  
 Move[] Attack = new Move[4];  
 Attack [0] = new Swagger();  
 Attack [1] = new ShadowBall();  
 Attack [2] = new Present();  
 Attack [3] = new Rest();  
 this.setMove(Attack);  
 }  
}  
  
class Rest extends StatusMove{  
 public Rest(){}  
 Effect Rest = new Effect().chance(0.3).turns(2).condition(Status.*SLEEP*);  
 protected void applySelfEffects(Pokemon att){att.addEffect(Rest); att.getHP();}  
 protected String describe(){return ("использует Rest");}

# Chansey.java

package com.company;  
import ru.ifmo.se.pokemon.\*;  
  
class Chansey extends Happiny{  
 public Chansey(String name, int level){  
 super(name, level);//формируем суперкласс для обращения к файлу в классе lab2  
 this.setStats(255,10,10,75,135,55);  
 this.setType(Type.*NORMAL*);  
 Move[] Attack = new Move[3];  
 Attack[0] = new Swagger();  
 Attack[1] = new ShadowBall();  
 Attack[2] = new Present();  
 this.setMove(Attack);  
 }  
}  
  
class Present extends PhysicalMove {  
 public Present() {  
 super(Type.*NORMAL*, 0, 90);  
 }  
  
 protected String describe() {  
 return ("использует Present");  
 }  
}

# Результат работы программы

Chansey Чхансей использует Swagger.

TapuFini Тапу Фини hits himself in confusion.

TapuFini Тапу Фини loses 3 hit points.

Chansey Чхансей использует Present.

TapuFini Тапу Фини loses 3 hit points.

TapuFini Тапу Фини использует BulkUp.

Chansey Чхансей использует Swagger.

TapuFini Тапу Фини misses

Chansey Чхансей использует Swagger.

TapuFini Тапу Фини использует ZenHeadbutt.

Chansey Чхансей loses 5 hit points.

Chansey Чхансей использует Swagger.

TapuFini Тапу Фини использует RockTomb.

Chansey Чхансей loses 4 hit points.

Chansey Чхансей использует Swagger.

TapuFini Тапу Фини использует BulkUp.

Chansey Чхансей использует Present.

TapuFini Тапу Фини loses 5 hit points.

TapuFini Тапу Фини использует BulkUp.

Chansey Чхансей использует Swagger.

TapuFini Тапу Фини misses

Chansey Чхансей использует Swagger.

TapuFini Тапу Фини hits himself in confusion.

TapuFini Тапу Фини loses 5 hit points.

TapuFini Тапу Фини faints.

Eevee Еевее from the team White enters the battle!

Eevee Еевее misses

Chansey Чхансей использует Present.

Eevee Еевее loses 5 hit points.

Eevee Еевее использует WorkUp.

Chansey Чхансей использует ShadowBall.

Eevee Еевее loses 1 hit points.

Eevee Еевее isn't affected by GHOST

Eevee Еевее misses

Chansey Чхансей использует Present.

Eevee Еевее loses 3 hit points.

Eevee Еевее использует WorkUp.

Chansey Чхансей использует ShadowBall.

Eevee Еевее loses 1 hit points.

Eevee Еевее isn't affected by GHOST

Eevee Еевее использует Facade.

Chansey Чхансей loses 6 hit points.

Chansey Чхансей использует Present.

Eevee Еевее loses 4 hit points.

Eevee Еевее misses

Chansey Чхансей использует Present.

Eevee Еевее loses 6 hit points.

Eevee Еевее faints.

Umbreon Умбрейон from the team White enters the battle!

Umbreon Умбрейон использует WorkUp.

Chansey Чхансей использует Swagger.

Umbreon Умбрейон hits himself in confusion.

Umbreon Умбрейон loses 4 hit points.

Chansey Чхансей использует Present.

Umbreon Умбрейон loses 5 hit points.

Umbreon Умбрейон hits himself in confusion.

Umbreon Умбрейон loses 5 hit points.

Chansey Чхансей использует ShadowBall.

Umbreon Умбрейон loses 3 hit points.

Umbreon Умбрейон hits himself in confusion.

Umbreon Умбрейон loses 6 hit points.

Umbreon Умбрейон faints.

Team White loses its last Pokemon.

The team Red wins the battle!

Вывод:

Я освоил ООП в языке джава